



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

**Grau en Enginyeria Mecànica**

**DESENVOLUPAMENT D'APLICACIONS DE REALITAT VIRTUAL PER  
ENTRENAMENT DE LES EXTREMITATS SUPERIORS**



**Memòria i Annexos**

<b>Autors:</b>	Daniel Gimeno Gómez Ivan González Albaladejo
<b>Director:</b>	Gil Serrancolí Masferrer
<b>Co-Director:</b>	Jordi Torner Ribe
<b>Convocatòria:</b>	Gener 2020



## Resum

En els darrers anys s'ha vist incrementat el nombre de notícies relacionades amb la realitat virtual, la gran majoria d'aplicacions que s'han desenvolupat per a aquesta tecnologia ha sigut per l'oci; però poc a poc han anat apareixent d'altres relacionades amb altres camps d'interès, i un a destacar és el de la biomèdica, que en poc temps ha aconseguit implementar varies aplicacions de realitat virtual en tot un conjunt de camps de la medicina, com la rehabilitació pediàtrica, la rehabilitació de lesions cerebrals traumàtiques, i la rehabilitació per post vessament cerebral, per anomenar-ne algunes.

Aquest treball s'emmarca en un projecte, en col·laboració amb l'Associació Diversitat Funcional d'Osona, per personalitzar i posar en marxa tractaments de rehabilitació basats en la utilització de la realitat virtual. La rehabilitació s'orienta a persones que han sofert ictus, ja que malauradament en són més de 13.000 les que ingressen als hospitals catalans cada any. Es va crear un exercici que consisteix en la realització d'un moviment d'abducció horitzontal i d'un joc que permet combinar moviments de flexió i abducció, basant-se en un ambient futurista.

Per a dur a terme aquest projecte es van utilitzar varies eines de treball: un programa informàtic anomenat Unity3D (utilitzat per a crear videojocs), el sensor de moviment Kinect de Microsoft, les ulleres de realitat virtual Vive de l'empresa HTC i un programa de creació i edició de bases de dades en l'àmbit informàtic, anomenat phpMyAdmin.

Aquesta aplicació, que s'ha provat en quatre persones sanes, és útil tant per pacients (podent dur a terme certs exercicis i jocs) com per professionals mèdics (aquests poden veure certs valors guardats mitjançant el sensor de moviment i així extreure les conclusions corresponents).

La finalitat d'aquest projecte és permetre que els exercicis de rehabilitació siguin més lúdics i, per tant, més amens, com també més efectius.

## Resumen

En los últimos años se ha visto incrementado el número de noticias relacionadas con la realidad virtual y la gran mayoría de aplicaciones que se han desarrollado para esta tecnología ha sido por ocio; pero poco a poco han ido apareciendo de otras relacionadas con otros campos de interés, y uno a destacar es el de la biomédica, que en poco tiempo ha conseguido implementar varias aplicaciones de realidad virtual en todo un conjunto de campos de la medicina, como la rehabilitación pediátrica, la rehabilitación de lesiones cerebrales traumáticas, y la rehabilitación por post derrame cerebral, para nombrar algunas.

Este trabajo se enmarca en un proyecto, en colaboración con la *Associació Diversitat Funcional d'Osona*, para personalizar y poner en marcha tratamientos de rehabilitación basados en la utilización de la realidad virtual. La rehabilitación se orienta a personas que han sufrido un ictus, ya que desgraciadamente son más de 13.000 las que ingresan a los hospitales catalanes cada año. Se creó un ejercicio que consiste en la realización de un movimiento de abducción horizontal y de un juego que permite combinar movimientos de flexión y abducción, basándose en un ambiente futurístico.

Para llevar a cabo este proyecto se utilizaron varias herramientas de trabajo: un programa informático llamado Unity3D (utilizado para crear videojuegos), el sensor de movimiento Kinect de Microsoft, las gafas de realidad virtual Vive de la empresa HTC y un programa de creación i edición de bases de datos en el ámbito informático, llamado phpMyAdmin.

Esta aplicación, que se ha probado con cuatro personas sanas, es útil tanto para pacientes (pudiendo llevar a cabo ciertos ejercicios y juegos) como para profesionales médicos (estos pueden ver ciertos valores guardados mediante el sensor de movimiento y así extraer las conclusiones correspondientes).

La finalidad del proyecto es permitir que los ejercicios de rehabilitación sean más lúdicos y, por lo tanto, más ameno, como también más efectivos.

## Abstract

In recent years, the number of news related about the virtual reality has increased, and the great part of the applications that it has been developing for this technology has been for pure leisure. But little by little are appearing other applications with other fields of interest, and the one to remark is the biomedical one, that in short time has got implement some applications of virtual reality in a set of medicine fields, like the paediatric rehabilitation, the traumatic brain injury rehabilitation, and the post stroke rehabilitation, to name a few.

This work is define in a project, in collaboration with the *Associació Diversitat Funcional d'Osona*, to personalize and to set in motion treatments of rehabilitation based in the use of virtual reality. The rehabilitation is oriented for those people who have suffered an ictus, because unfortunately there are more than 13.000 those who admit in the Catalan hospitals every year. It has created an exercise that consist in the realization of a horizontal abduction movement and a game that allows combining flection and abduction movements, basing on a futuristic atmosphere.

For realising the project, it was use a few work tools: a computer program called Unity3D (it is used to create videogames), the movement sensor Kinect from Microsoft, the Vive virtual reality glasses from the HTC manufacturer and a program of creation and edition of data bases in the informatics world, called phpMyAdmin.

This application, that it has checked with four healthy people, is useful as for patients (even it is possible to do some exercises and games) as for medical professionals (they can see certain kept values by the movement sensor and thus extract the pertinent conclusions).

The purpose of the project is to allow that the rehabilitation exercises would be more playful and, therefore, more enjoyable as more effective too.



## Agraïments

Durant el temps que hem estat realitzant aquest treball ens hem creuat i trobat amb gent que ens ha fet el camí una mica menys tortuós, ja sigui en l'àmbit estudiantil com en l'àmbit social, és per això que els hi volem agrair la seva dedicació, paciència i enteniment.

És per això que volem agrair la nostra universitat, la Universitat Politècnica de Catalunya (UPC), a l'Associació Diversitat Funcional d'Osona (ADFO), la fisioterapeuta Cristina Molas, al nostre professorat, en especial al Gil Serrancolí, Jordi Torner i Francesc Alpiste, per ajudar-nos en moments crítics i simplificar-nos certes situacions. Tampoc ens oblidem de l'Elodie Medina i en Víctor Rofes Pujol, antics alumnes que van realitzar el projecte el quadrimestre anterior, i van ser allà per aclarir-nos certes qüestions sobre el treball. Per altra banda ens agradaria esmentar als familiars i amics que ens han ajudat en aquest camí, que ens han fet desconnectar en certs moments quan era necessari, ens han ajudat a treballar més dur en els moments més difícils i sentir-se i fer-nos sentir orgullosos del nostre treball.

Gràcies per fer-ho possible.





## Glossari

ADFO	Associació Diversitat Funcional d'Osona
<i>Canvas</i>	L'àrea on han d'estar els elements de la interfície de l'usuari
DNI	Document Nacional d'Identitat
<i>Frame</i>	Cadascuna de les imatges estàtiques que formen part de la successió d'imatges que formen una animació, i que a la vista produeixen sensació de moviment.
Kinect	Sensor de moviment de l'empresa Microsoft
RV	Realitat Virtual
TFG	Treball de Final de Grau
UPC	Universitat Politècnica de Catalunya



# Índex

<b>RESUM</b>	<b>II</b>
<b>RESUMEN</b>	<b>III</b>
<b>ABSTRACT</b>	<b>IV</b>
<b>AGRAÏMENTS</b>	<b>VI</b>
<b>GLOSSARI</b>	<b>VIII</b>
<b>1. PREFACI</b>	<b>1</b>
1.1. Origen del treball .....	1
1.2. Motivació .....	1
1.3. Requeriments previs .....	2
<b>2. INTRODUCCIÓ</b>	<b>3</b>
2.1. Objectius del treball .....	3
2.2. Abast del treball .....	3
<b>3. ESTAT DE L'ART</b>	<b>5</b>
<b>4. EINES EMPRADES</b>	<b>11</b>
4.1 Unity .....	11
4.1.1 Requisits per poder utilitzar Unity .....	12
4.1.2 La interfície de Unity.....	12
4.2 Llenguatges de programació .....	13
4.3 XAMPP i phpMyAdmin.....	14
4.4 Kinect.....	16
4.4.1 Requisits per poder utilitzar el Kinect .....	17
4.4.2 Especificacions tècniques .....	17
4.4.3 Kinect per a Windows SDK 2.0 .....	19
4.5 Ulleres HTC Vive .....	20
4.5.1 Requisits per poder utilitzar les HTC Vive .....	20
4.5.2 Hardware de les HTC Vive .....	20
4.5.3 Software necessari pel funcionament del hardware.....	21
<b>5. ESTRUCTURA I FUNCIONALITATS DE L'APLICACIÓ</b>	<b>23</b>

5.1	Abducció horitzontal.....	24
5.2	Flexió braços (desactivació bomba).....	26
5.3	Càlcul d'angles.....	27
5.3.1	Angle d'abducció.....	28
5.3.2	Angle de flexió.....	29
<b>6.</b>	<b>METODOLOGIA DEL DISSENY DE L'APLICACIÓ .....</b>	<b>30</b>
6.1	Abducció horitzontal.....	30
6.2	Flexió braços (desactivació bomba).....	38
<b>7.</b>	<b>PLANIFICACIÓ DEL PROJECTE .....</b>	<b>48</b>
7.1	Pla de treball .....	48
7.2	Diagrama de Gantt.....	51
7.3	Requeriments funcionals i no funcionals .....	52
<b>8.</b>	<b>ASSAJOS AMB VOLUNTARIS .....</b>	<b>54</b>
8.1	Abducció Horitzontal .....	55
	Voluntari 1.....	55
	Voluntari 2.....	59
	Voluntari 3.....	61
	Voluntari 4.....	64
8.2	Flexió braços (desactivació bomba).....	66
	Voluntari 1.....	66
	Voluntari 2.....	69
	Voluntari 3.....	71
	Voluntari 4.....	71
<b>9.</b>	<b>ANÀLISI DE L'IMPACTE AMBIENTAL .....</b>	<b>74</b>
<b>10.</b>	<b>CONCLUSIONS .....</b>	<b>76</b>
<b>11.</b>	<b>LÍNIES FUTURES .....</b>	<b>77</b>
<b>12.</b>	<b>PRESSUPOST I ANÀLISIS ECONÒMICA .....</b>	<b>78</b>
12.1	Despeses de hardware.....	78
12.2	Despeses de software .....	79
12.3	Despeses del consum elèctric.....	79
12.4	Despeses de personal .....	80
12.5	Resum de despeses.....	81

<b>13. BIBLIOGRAFIA</b>	<b>82</b>
<b>ANNEX 1. INSTRUCCIONS PER INICIALITZAR EL JOC</b>	<b>86</b>
A1.1 Connectar el sensor Kinect.....	86
A1.2 Connectar les ulleres HTC Vive.....	87
A1.3 Iniciar la base de dades .....	87
<b>ANNEX 2. ADJUNT DE L'APLICACIÓ</b>	<b>88</b>
<b>ANNEX 3. CODIS</b>	<b>89</b>
A3.1 Codis C# .....	89
Actions .....	89
Angulos .....	92
BodySourceManager .....	94
BodySourceUpperBody.....	96
bomba .....	134
Button_sound .....	140
CanviaEscena.....	141
CopiaDNI .....	142
DataInserter .....	143
DeletePatient .....	145
DestroyOnload .....	148
Envioangulos .....	149
explosion .....	153
Listado .....	154
Login .....	155
LoginPaciente.....	157
mostrardias .....	158
NameController .....	159
neutralitzada .....	160
PatientInsert.....	161
pushbutton.....	162
RefreshRemoto .....	163
SalirScena .....	164
SampleButton .....	165
ShopScrollList .....	166
ShopScrollListDias .....	169
SipleObjectPool.....	172

Vector4 .....	175
A3.2 Codis PHP .....	176
AngleInsert .....	176
DeletePatient .....	178
InsertPatient.....	179
InsertUser .....	180
listangulos .....	181
Login .....	182
LoginPaciente.....	183
mostrardias .....	184
SeleccionDoctor .....	185

## Índex de figures

Figura 3.1 a Equipament grup experimental. b Equipament grup de control.....	6
Figura 3.2 Equipament grup 2D.....	7
Figura 3.3 Equipament grup 3D.....	7
Figura 3.4. Sistema del Wii Fit [15].....	9
Figura 3.5 Aplicació tele-rehabilitació .....	10
Figura 4.1. Eines utilitzades .....	11
Figura 4.2 Interfície de Unity.....	13
Figura 4.3 Pantalla d'inici XAMPP.....	14
Figura 4.4 Menú principal phpMyAdmin .....	15
Figura 4.5 Creació nova taula .....	15
Figura 4.6 Creació nova columna .....	16
Figura 4.7. Kinect bat un Rècord Guinness el 2011 [26] .....	17
Figura 4.8 Punts detectats per Kinect [30] .....	19
Figura 4.9 Pantalla Microsoft Kinect Studio .....	19
Figura 4.10 Equipament de RV HTC Vive [33] .....	20
Figura 4.11. Pestanya informativa del Steam VR .....	21
Figura 4.12. Indicació de no connectat d'una de les estacions base .....	22
Figura 4.13. Indicació de bateria d'un dels controladors .....	22
Figura 5.1 Estructura aplicació .....	23
Figura 5.2 Exercici d'abducció horitzontal [35] .....	24
Figura 5.3 Canvas abducció horitzontal .....	25
Figura 5.4 Esquema desactivació bomba .....	26
Figura 5.5 Angle de flexió [36].....	27
Figura 5.6 Representació pla frontal [37].....	28
Figura 5.7 Representació pla sagital [37] .....	29
Figura 6.1. Menú de selecció d'exercicis .....	30
Figura 6.2 Accions al clicar el botó d'abducció horitzontal.....	30
Figura 6.3 Canvas abducció horitzontal .....	31
Figura 6.4 Programa "HABDUP" .....	31
Figura 6.5 Corutina "SeemaforoHABDUP" .....	32
Figura 6.6 "HABDUptime" dins d'"Update" .....	32
Figura 6.7 Condició " if (HABDUP == true)" .....	33
Figura 6.8 Corutina "TemporizadorHABDUP" .....	34
Figura 6.9 Funció "excelHABDUP" .....	35
Figura 6.10 Funcionalitat botó "Atrás".....	36
Figura 6.11 Funció "envioHABDUP" .....	37
Figura 6.12 Corutina "AngleInsert" .....	37
Figura 6.13 Script "AngleInsert.php" .....	38

Figura 6.14 Funcionalitats botó joc desactivació bomba .....	39
Figura 6.15 Funció “canviaBomba” i “playBomb” .....	39
Figura 6.16 Variables a “Update” .....	39
Figura 6.17 Variables i condicions dins d’“Update” .....	40
Figura 6.18 Condició “Nivell == 0” .....	41
Figura 6.19 Colliders mans i polsadors.....	41
Figura 6.20 Script “pushbutton” .....	42
Figura 6.21 Condició “pushbutton.numero == 2” .....	42
Figura 6.22 Corutina “Temporizador” i Figura XX Funcio “BombDesactivation” .....	43
Figura 6.23 Condició “Nivell == 1” .....	43
Figura 6.24 Condició “countdown <= 0” .....	43
Figura 6.25 Funció “Explode” .....	44
Figura 6.26 Corutina “TempGameOver” .....	44
Figura 6.27 Canvas “GameOver” .....	45
Figura 6.28 Funció “backMenu” i funció “quitbomb” .....	45
Figura 6.29 Funció “envioBomba” .....	46
Figura 6.30 Condició “Nivell == 8” .....	46
Figura 6.31 Canvas “Win” .....	47
Figura 7.1 Metodologia per a la realització del treball .....	50
Figura 8.1 Abducció voluntari 1 (abducció horitzontal).....	57
Figura 8.2 Flexió voluntari 1 (abducció horitzontal).....	57
Figura 8.3 Abducció voluntari 2 (abducció horitzontal).....	59
Figura 8.4 Flexió voluntari 2 (abducció horitzontal).....	59
Figura 8.5 Abducció voluntari 3 (abducció horitzontal).....	61
Figura 8.6 Flexió voluntari 3 (abducció horitzontal).....	62
Figura 8.7 Abducció voluntari 4 (abducció horitzontal).....	64
Figura 8.8 Flexió voluntari 4 (abducció horitzontal).....	64
Figura 8.9 Abducció voluntari 1 (bomba).....	66
Figura 8.10 Flexió voluntari 1 (bomba) .....	67
Figura 8.11 Abducció voluntari 2 (bomba).....	69
Figura 8.12 Flexió voluntari 2 (bomba) .....	69
Figura 8.13 Abducció voluntari 4 (bomba).....	71
Figura 8.14 Flexió voluntari 4 (bomba) .....	72
Figura 9.1. Símbol per marcar els aparells elèctrics o electrònics [41].....	75
Figura A1. 1 Esquema de connexió del sensor Kinect a l'ordinador i al corrent elèctric [33] .....	86
Figura A1. 2. Col·locació del sensor Kinect [34] .....	86
Figura A1. 3 Activació dels mòduls Apache i MySQL .....	87



## Índex de taules

Taula 4.1 Requisits Kinect.....	17
Taula 4.2 Característiques tècniques Kinect .....	18
Taula 4.3 Requisits HTC Vive .....	20
Taula 7.1 Requeriments funcionals i no funcionals .....	53
Taula 8.1. Informació general dels voluntaris .....	54
Taula 8.2 Dades estadístiques voluntari 1 (abducció horitzontal).....	58
Taula 8.3 Dades estadístiques voluntari 2 (abducció horitzontal).....	60
Taula 8.4 Dades estadístiques voluntari 3 (abducció horitzontal).....	63
Taula 8.5 Dades estadístiques voluntari 4 (abducció horitzontal).....	65
Taula 8.6 Dades estadístiques voluntari 1 (bomba).....	68
Taula 8.7 Dades estadístiques voluntari 2 (bomba).....	70
Taula 8.8 Dades estadístiques voluntari 4 (bomba).....	73
Taula 12.1. Costs de hardware .....	78
Taula 12.2. Costs de software .....	79
Taula 12.3. Cost del consum elèctric.....	80
Taula 12.4. Sou dels treballadors .....	80
Taula 12.5. Formació dels treballadors .....	80
Taula 12.6. Cost final del projecte .....	81



## **1. Prefaci**

En el pas del temps, la medicina ha anat millorant les seves tècniques i ha pogut ampliar els seus camps de coneixement gràcies als avenços tecnològics. Entre ells, cal anomenar la realitat virtual, que s'ha anat fent pas en l'àmbit de la rehabilitació física i cognitiva. Això es deu, en part, perquè és un tipus de tractament no invasiu que permet la creació artificial de diferents sensacions sensorials que el pacient pot percebre com a reals. Tot i que pugui semblar mentida, la rehabilitació física emprant la realitat virtual és un camp molt poc examinat i que, per tant, dona lloc a moltes futures millores.

### **1.1. Origen del treball**

Aquest treball és la continuació de certs Treballs de Fi de Grau, els quals van connectar el sensor Kinect amb el motor de videojocs Unity3D. Gràcies a això, es va aconseguir certa informació sobre els angles en unes parts concretes del cos mentre es realitzaven certs moviments. A continuació, es va aprofundir més i es va ampliar la bateria d'exercicis per als pacients i l'allotjament d'uns quants valors a una base de dades externa anomenada phpMyAdmin.

Seguint les recomanacions i condicions inicials que va requerir l'Associació Diversitat Funcional d'Osona (ADFO) i amb el material facilitat per l'empresa Visyon360, es va procedir a la realització d'aquest projecte.

### **1.2. Motivació**

La motivació per a endinsar-se en aquest projecte va ser, per una banda, la de poder ajudar a les persones amb una discapacitat física. Ajudar a dur a terme alguns exercicis, que per experiència pròpia es sap que són molt pesats i, d'aquesta manera, fer-los sentir més contents i dur una vida una mica més còmode, encara que sigui de manera inapreciable per als demés.

Per altra banda, també va ser la de poder ajudar als fisioterapeutes a fer millor i més fàcilment el seu treball, és a dir, ajudar als pacients a recuperar-se d'una manera més ràpida i eficient.

I per últim, era una motivació la idea de treballar amb una tecnologia realment innovadora i en desenvolupament com és la RV, i així ajudar-nos també a nosaltres a satisfer les nostres inquietuds i intentar millorar, en certa manera, la visió sobre aquesta tecnologia que és molt innovadora però, per altra banda, molt desconeguda per a una gran part de la població.



### 1.3. Requeriments previs

Per al correcte desenvolupament de l'aplicació que es crea en aquest projecte, es necessita una sèrie de coneixements; en primer lloc, tenir un bon domini del motor de videojocs Unity3D, que a la vegada comporta que l'usuari tingui un bon nivell del llenguatge de programació C#, que és l'utilitzat per poder programar els scripts fonamentals per al projecte de Unity3D. Per altra banda, cal destacar també la necessitat de conèixer MySQL, per a la creació i edició de bases de dades virtuals i del llenguatge de programació PHP, que és l'eina que permet connectar les dades obtingudes i tractades en els scripts de C# per a que arribin a la base de dades creada.

A més a més, també cal destacar el material necessari que s'ha emprat:

- Un ordinador amb sistema operatiu Windows 7 o superior, de 64 bits
- El motor de videojocs Unity3D
- El programari de Visual Studio
- El sensor de moviment Kinect
- Un adaptador que permeti connectar el Kinect a l'ordinador
- El programari Microsoft Kinect for Windows SDK
- Les ulleres de RV HTC Vive
- El software Steam VR

## **2. Introducció**

### **2.1. Objectius del treball**

Els objectius del treball sempre han estat en benefici tant del pacient com del professional en qüestió.

Respecte al pacient, es vol aconseguir que el seu procés de rehabilitació diària sigui una mica menys monòton i, per tant, que els pacients tinguin una millor predisposició a fer els exercicis corresponents i així obtenir uns resultats més efectius.

Pel que fa als fisioterapeutes, la finalitat que es busca és ajudar a fer el seu treball més fàcil, mitjançant les dades de l'historial que poden ser útils després de ser analitzades i, evidentment, si els pacients van amb una millor predisposició també s'estarà fent que la seva labor sigui més avinent.

Així doncs, l'objectiu principal d'aquest projecte és actualitzar l'aplicació de realitat virtual, que es porta fent en diversos TFG, augmentant la bateria d'exercicis per les extremitats superiors del cos. Aquestes activitats s'han realitzat seguint els consells i les demandes d'una fisioterapeuta professional de l'ADFO. Es va procedir a elaborar dues noves aplicacions, una on l'usuari realitza l'exercici d'abducció horitzontal dels braços i, per altra banda, es va desenvolupar un joc, que busca un entorn més lúdic i vistós que qualsevol altre exercici, que consisteix en la flexió dels braços.

Però per a poder realitzar i complir aquests objectius, se n'ha plantejat un altre de necessari per una correcta execució: exportar les dades útils per als fisioterapeutes, que serien les posicions de les articulacions de les extremitats superiors, i els angles de flexió i abducció.

### **2.2. Abast del treball**

El projecte en qüestió és la continuació d'un seguit de treballs anteriors, els quals han facilitat la creació dels nous exercicis de realitat virtual i s'ha reaprofitat la base de dades on s'exportaran les noves extretes en aquest treball.

Per tant, l'abast d'aquest projecte es centra en la creació de noves escenes (tant exercicis com jocs) i l'exportació de diferents dades que ja es tenien d'aquestes.

Tant els tutors com la fisioterapeuta varen recomanar exportar no només un sol angle, sinó la posició dels punts de les extremitats superiors que recull el Kinect. Davant d'aquest repte, es va decidir extreure

les coordenades d'aquests punts a un entorn de Microsoft Excel, el qual permet interpretar i entendre els resultats obtinguts d'una manera més visual i còmode.

Per altra banda, es va decidir no desemparar la base de dades que es tenia anteriorment; per aquesta raó, s'han exportat algunes dades que s'han trobat més significatives i que poden ser de més ajut.

### **3. Estat de l'art**

En els últims anys, les aplicacions de RV s'han desenvolupat en diferents àmbits de la medicina: rehabilitació post-ictus [1], rehabilitació pediàtrica [2], rehabilitació de la marxa [3], rehabilitació de lesions cerebrals traumàtiques [4], salut mental [5] i, tot i que no hi ha unanimitat en quan a l'efectivitat del seu ús, s'aprecia la voluntat de continuar valorant i generalitzant-ne l'ús. En general, s'afirma que els entorns de realitat virtual milloren el goig i la motivació en les tasques de rehabilitació [1].

Pel que fa a la immersió, és ben acceptat que els programes de rehabilitació d'ictus siguin més eficaços quan es realitza en un entorn immersiu que fomenti un compromís de major durada [6]. A més, un recent meta-anàlisi [7] va suggerir que les intervencions de RV són, com a mínim, tan efectives com la fisioteràpia convencional per millorar els resultats de les extremitats inferiors en pacients que han patit un ictus.

Els assajos que utilitzen intervencions de RV per millorar els resultats de les extremitats inferiors en pacients adults que han patit ictus es centren en l'entrenament de l'equilibri [8], entrenament del control postural [9] i control de la marxa [10].

En altres estudis on es registren paràmetres de marxa espai-temporals de velocitat de la marxa, cadència, longitud de pas, temps de pas i longitud de pas, s'han detectat angles articulars de les extremitats superiors i inferiors per demostrar una millora funcional provocada per la intervenció de la realitat virtual. Inclús es van trobar diferències significatives a favor del grup de RV per l'equilibri funcional, la velocitat de la marxa, la cadència i la longitud del pas dins la població d'ictus.

Per altra banda, alguns autors avalen l'eficàcia de la RV en la rehabilitació després d'un ictus, ja que milloren la motivació dels pacients degut a que els exercicis poden ser adaptats [11].

A continuació, s'ha donat èmfasi a un seguit d'estudis que aprofundeixen en certes aplicacions de realitat virtual d'un ús fisioterapèutic.

La facultat de Medicina d'Ottawa va escriure un article el 2017 que tracta precisament sobre la rehabilitació a distància amb l'ajuda de les eines que aporta la realitat virtual (també anomenada tele-rehabilitació) per a persones que han sofert un ictus [12].

L'article descriu l'experiment que varen portar a terme 40 individus. Cal destacar que són persones que poden estar dempeus almenys 2 minuts i, a més, provenen d'un procés de rehabilitació, ja finalitzat, del seu hospital corresponent. Aquestes 40 persones són dividides aleatòriament en 2 grups.

El primer dels grups, l'experimental, utilitza la realitat virtual mitjançant el sensor de moviment Kinect. Els participants realitzen exercicis per millorar l'estabilitat, per simular els passos a l'hora de caminar, per executar certs moviments amb els braços que simulen l'assoliment d'objectes, per enfortir la musculatura i millorar la forma física aeròbica (de forma moderada). El segon grup, el de control, treballava amb una tauleta, la qual tenia unes aplicacions prèviament seleccionades per a millorar la cognició, la motricitat de les mans i el rastreig o seguiment visual.



*Figura 3.1 a Equipament grup experimental. b Equipament grup de control.*

Ambdós grups van realitzar 30 minuts d'aquests exercicis, cinc cops a la setmana durant sis setmanes, el que comportava un total de 900 minuts (15 hores).

Cal destacar que la dificultat i intensitat dels exercicis realitzats per al grup experimental eren graduats de forma remota.

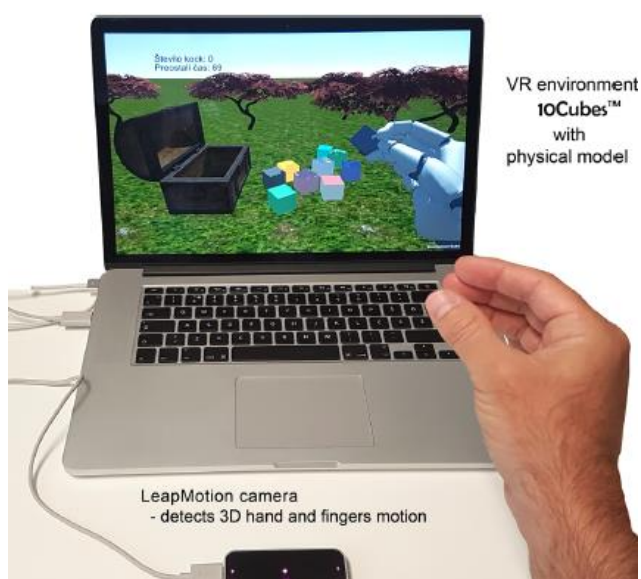
L'article conclou que la realitat virtual té i pot arribar a tenir eines molt útils i potents per a poder millorar el camp de la rehabilitació i obre una porta molt interessant a la tele-rehabilitació. La tele-rehabilitació permet, en primer lloc, estalviar temps i costos, ja que si el pacient viu a un lloc allunyat del centre mèdic corresponent, o no es pot desplaçar de casa i el professional en qüestió ha d'anar a casa del pacient a dur a terme el tractament es perd molt temps. En segon lloc, també permet complementació i flexibilitat als pacients, ja que normalment en un procés de rehabilitació els pacients han de fer exercicis a casa, i la realitat virtual permet fer exercicis que amb les eines de casa no es podrien realitzar. Per altra banda, permet fer les activitats quan el pacient ho desitgi, en comptes d'haver d'anar a un centre mèdic. Per tant, és més compatible a l'hora de treballar o complementar-ho amb qualsevol activitat. En tercer lloc, aporta una motivació extra als pacients. Són activitats més interactives i lúdiques que permeten al pacient fins i tot divertir-se mentre realitza els exercicis corresponents (també en el cas del sensor Kinect, que és una eina fàcil d'instal·lar i d'utilitzar).

Com a últim i més important, menciona que no causa efectes adversos al pacient, encara que per altra banda no hi ha estudis suficients per a demostrar que és una teràpia efectiva i és necessari investigar més en aquest àmbit.



En un altre estudi més recent (2019), escrit per Imre Cikajlo i Karmen Peterlin Potisk [13], es va treballar amb pacients amb Parkinson que utilitzaven eines de realitat virtual 3D.

En aquest cas van comptar amb 20 pacients, separats en dos grups. El primer d'ells treballava amb les ulleres 3D Oculus Rift CV1 i el segon amb un ordinador portàtil. Realitzaven els mateixos exercicis però en un cas utilitzaven les ulleres de realitat virtual i en l'altre cas treballaven en un àmbit de dos dimensions. En aquest estudi, els pacients treballaven amb un joc que van crear amb Unity3D; treballaven amb un sensor de moviment únicament per a la mà, el qual és, bàsicament, una càmera 4D (està formada per dues càmeres i tres sensors d'infrarojos). El joc consistia en recollir deu cubs de colors i pesos diferents i introduir-los dins d'una caixa més gran.



*Figura 3.2 Equipament grup 2D*



*Figura 3.3 Equipament grup 3D*

L'article mencionat afirma que els pacients que van utilitzar la tecnologia 3D van tenir un major interès i motivació en les activitats i que això va portar a uns millors resultats i eficàcia. Per altra banda, el grup que va treballar amb la pantalla es va sentir més relaxat i, gràcies a això, van tenir menys errors que el grup de 3D. Com a conseqüència, amb el temps van perdre l'interès i hi van posar menys esforç (al final de cada sessió se'ls efectuava vuit preguntes, una de les quals era si havien hagut de fer molt d'esforç per jugar; el rang de puntuació per pregunta anava de 2 a 14 punts).

Com a conclusió s'extreu que s'ha de seguir investigant amb aquestes tecnologies i que pot ser que la tecnologia de realitat virtual immersiva no sigui apte per a tots els pacients amb Parkinson, en particular amb aquells que porten ulleres, que han presentat problemes amb l'ajust de les ulleres 3D, molèsties visuals, cansament ocular i inclús miopia transitòria.

En un altre article de recerca biomèdica [14] es parla també de pacients amb Parkinson, els quals duen a terme tele-rehabilitació per a millorar la inestabilitat corporal mitjançant la taula d'exercicis Nintendo Wii Fit.

En aquest cas, es van seleccionar 76 pacients amb Parkinson separats en dos grups de 38 persones cadascun. El primer grup treballava a casa seva amb el Wii Fit i el segon efectuava un entrenament d'equilibri i integració sensorial en una clínica. Es van dur a terme 21 sessions de 50 minuts cadascuna, 3 dies a la setmana durant 7 setmanes consecutives. És important destacar que els pacients es van avaluar abans de començar el tractament, durant, al finalitzar-lo i al cap d'un mes d'haver-lo acabat.

L'equip del Wii Fit de l'estudi prèviament mencionat constava en primer lloc de la consola Wii, amb el seu corresponent sensor i comandament, la taula on s'hi puja, que és el mateix Wii Fit i, òbviament, el joc. A més a més, un ordinador amb una càmera permetia al fisioterapeuta realitzar un control visual via Skype.

D'aquest estudi s'extreu com a conclusió que en els dos tractaments els pacients milloren. Referent als que van dur a terme la tele-rehabilitació, van millorar més que l'altre grup en control postural, tant estàtic com dinàmic i, per altra banda, els que van realitzar el tractament en el centre mèdic van millorar més en mobilitat i equilibri dinàmic, tot i que les diferències entre els dos grups van ser insignificants.

Cal destacar que el sistema de Wii Fit és molt més barat i aporta més flexibilitat horària al pacient, és a dir, no té la necessitat d'anar a l'hospital a realitzar les activitats de rehabilitació i perdre el temps en anar i tornar de la consulta mèdica, així com també poder-les realitzar en qualsevol moment del dia quan el pacient més li convingui. Gràcies a aquests fets s'afirma que la realitat virtual, en aquest cas utilitzant el sistema Wii Fit, és adequat per a pacients amb Parkinson que tinguin un cuidador.



Figura 3.4. Sistema del Wii Fit [15]

Finalment, en un article més antic però no per això menys interessant [16], es considera un estudi sobre tele-rehabilitació per al turmell i peu per a pacients que han patit un ictus. Tracta d'un prototip per al turmell connectat a un ordinador, el qual està connectat amb un terapeuta extern via internet.

Durant les quatre setmanes que durava l'estudi es van realitzar quatre exercicis diferents:

- En la setmana 1, exercici de pilotatge d'avió.
- La setmana 2, pilotatge d'avions amb efectes hàptics.
- La setmana 3, navegació en vela i pilotatge d'avions amb efectes hàptics.
- L'última setmana, la quarta, navegació en vela i pilotatge d'avions amb efectes hàptics realitzats en tele-rehabilitació.

Aquests exercicis constaven d'una hora, tres cops a la setmana durant les quatre setmanes ja esmentades. Cal destacar que durant les primeres tres setmanes, el terapeuta es trobava a la mateixa habitació que el pacient i que a l'última setmana el terapeuta era en un altre lloc, fent uns seguiment mitjançant càmeres.

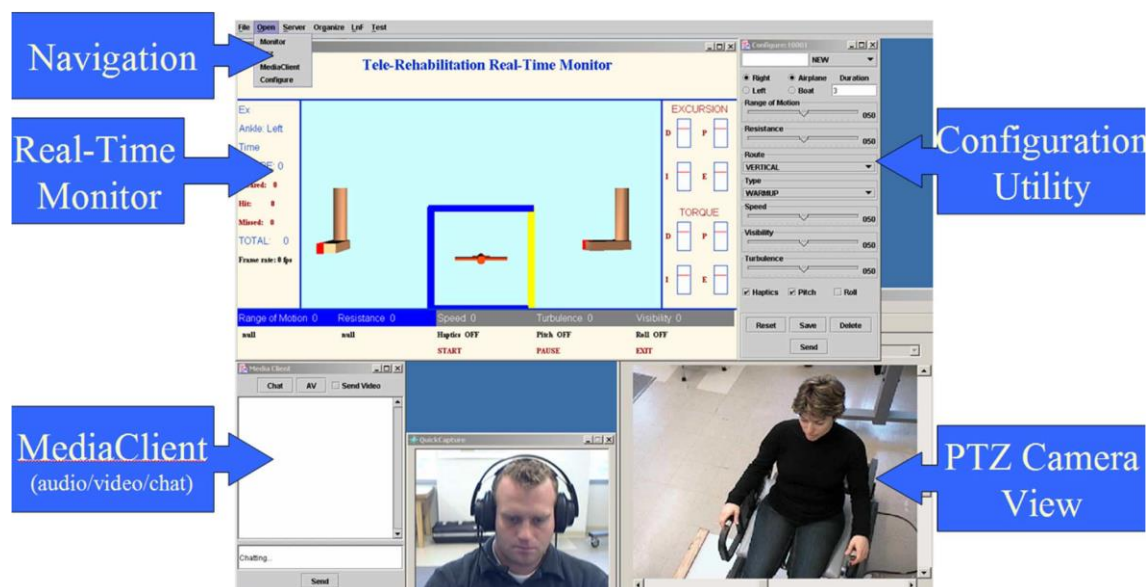


Figura 3.5 Aplicació tele-rehabilitació

Entre els valors que rep el terapeuta es trobaven: la posició i l'orientació del pla, la posició del peu del subjecte, les mesures de la força a la plataforma i tots els paràmetres de simulació.

En aquest estudi s'afirma que els pacients van passar d'un 70% de precisió a l'inici a un 83% a l'última setmana; en termes de potència de 0.44 W a 0.71 W; pel que fa a les repeticions, es va passar d'una mitjana de 173 a 473 repeticions per setmana; referent a la duració de 32 a 50 minuts i per últim, respecte l'eficiència de 56% a 72%. És a dir, els pacients van millorar en tots els aspectes.

Els resultats indiquen que el rendiment del pacient no es va veure afectat per la localització del terapeuta (situat a la mateixa habitació que el pacient o a un lloc "remot").

Com a conclusions s'extreu que s'haurien de realitzar proves addicionals del sistema de rehabilitació integrada de realitat virtual en condicions de xarxa dolentes, així com quan els pacients haurien de realitzar els exercicis de forma independent, sense que hi hagi cap terapeuta a la sala. La principal conclusió aquí pel que fa a tele-rehabilitació és que el rendiment del grup es va mantenir durant la transició de la rehabilitació *in situ* a remota.

## 4. Eines emprades

Com en tot projecte, es necessita d'una sèrie de programes i maquinari necessari per al correcte desenvolupament d'aquest.

A la Figura 4.1, l'eina "mare" del projecte és el motor de videojocs Unity, que envia informació tant al paquet de programari lliure XAMPP com a les ulleres de RV HTC Vive; a XAMPP li envia totes les dades registrades durant el joc (angles, temps, nivell, ...), que a la vegada les envia a la base de dades phpMyAdmin. Les ulleres de RV permeten a l'usuari visualitzar tot l'entorn virtual, creat amb el Unity, de manera immersiva. Per altra banda, el propi Unity també rep informació de phpMyAdmin a través de XAMPP (totes les dades que havia enviat anteriorment també es poden veure a l'aplicació), de les HTC Vive (envien els moviments del cap per a que l'avatar mogui igual que l'usuari) i del sensor de moviment Kinect (envia tots els moviments de l'usuari per a que posteriorment siguin reproduïts a l'avatar).

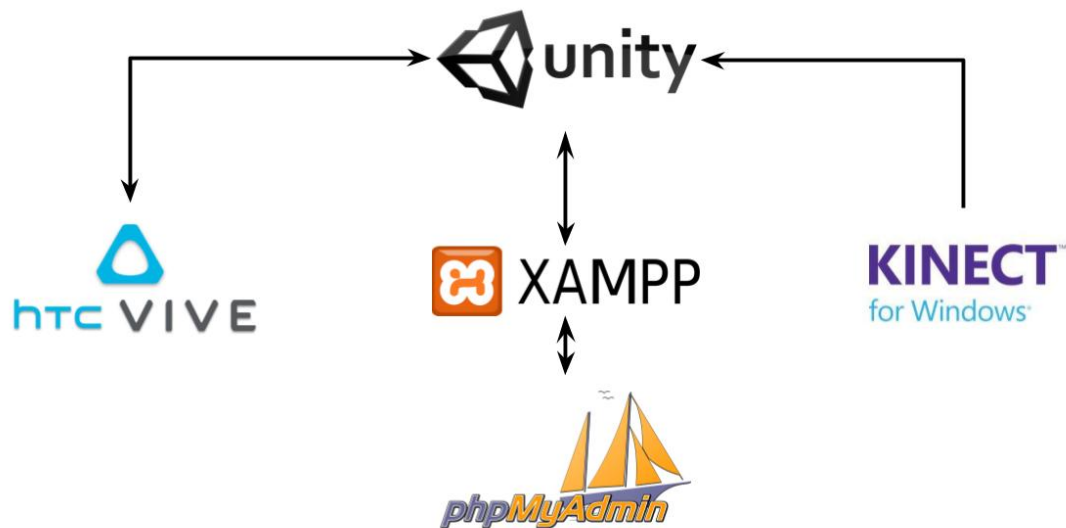


Figura 4.1. Eines utilitzades

### 4.1 Unity

Unity és un motor de videojoc multi plataforma desenvolupat per l'empresa Unity Technologies, fundada el 1988 per David Helgason, Nicholas Francis i Joachim Ante a Copenhagen, Dinamarca.

Es troba disponible pels sistemes operatius de Microsoft Windows, Mac OS i Linux. A més, permet crear aplicacions en múltiples plataformes, com poden ésser iOS, Android, Samsung Smart TV, Play Station 4, Play Station Vita, Xbox One, Wii U, Google Cardboard, per anomenar-ne unes quantes.

Cal destacar també que Unity ofereix diverses llicències adequades als ingressos de cada empresa. Així doncs, es troba la Unity Personal (totalment gratuïta, la que s'ha utilitzat en aquest treball), per aquells negocis que tinguin uns ingressos iguals o inferiors als 100.000\$ anuals; després proposa la Unity Plus (pagament de 35\$ al mes, amb un període de compromís d'un any) per aquelles que els seus ingressos no superin els 200.000\$ anuals; i la Unity Pro (amb un preu de 125\$ al mes), que la pot descarregar qualsevol, sense tenir en compte els ingressos i que dona accés a tots els serveis de Unity.

#### 4.1.1 Requisits per poder utilitzar Unity

En aquest apartat es mostren els requisits mínims que ha de tenir l'ordinador per a la correcta utilització de Unity[17]:

- Sistema operatiu: Windows 7 SP1+, 8 i 10 (només en versions de 64 bits); macOS 10.11+; Linux Ubuntu 16.04, la 18.04 i la CentOS 7.
- CPU: pel suport del conjunt d'instruccions és necessari posseir el SSE2 (Streaming "Single Instruction Data" Extensions 2)
- GPU: targeta gràfica DirectX 10 (shader model 4.0)

#### 4.1.2 La interfície de Unity

La interfície de Unity es compon de varies finestres que el propi usuari pot agrupar, minimitzar i personalitzar al seu gust. D'aquesta manera l'editor pot ésser diferent d'un projecte a un altre, depenent de la preferència personal o del tipus de treball que s'estigui realitzant[18].

La primera que cal destacar és la finestra del projecte (requadre verd de la Figura 4.2 ), situada a la cantonada inferior esquerra.

Just al centre de la pantalla es troba la vista d'escena (requadre lila de la Figura 4.2). En ella es permet visualitzar i editar l'escena del joc, així com també veure el propi joc en funcionament.

A l'esquerra es troba la finestra de la jerarquia (requadre groc de la Figura 4.2). Aquí es pot trobar la jerarquia de cada objecte en l'escena.

Al lateral dret de la pantalla, es troba la finestra de l'inspector (requadre blau de la Figura 4.2), que permet l'edició i la visualització de totes les propietats de l'objecte seleccionat en la finestra de la jerarquia.

Finalment, a la part superior de la pantalla es situa la barra d'eines (requadre vermell de la Figura 4.2), que dona accés a totes les eines del programa. A més, també es troba els controls de reproducció, pausa i passos, per quan es vol posar en marxa el joc que s'està creant. Aquesta és la única finestra no editable,

ja que, de fet, no es pot considerar com a una finestra, sinó com a una part necessària de la interfície de Unity.

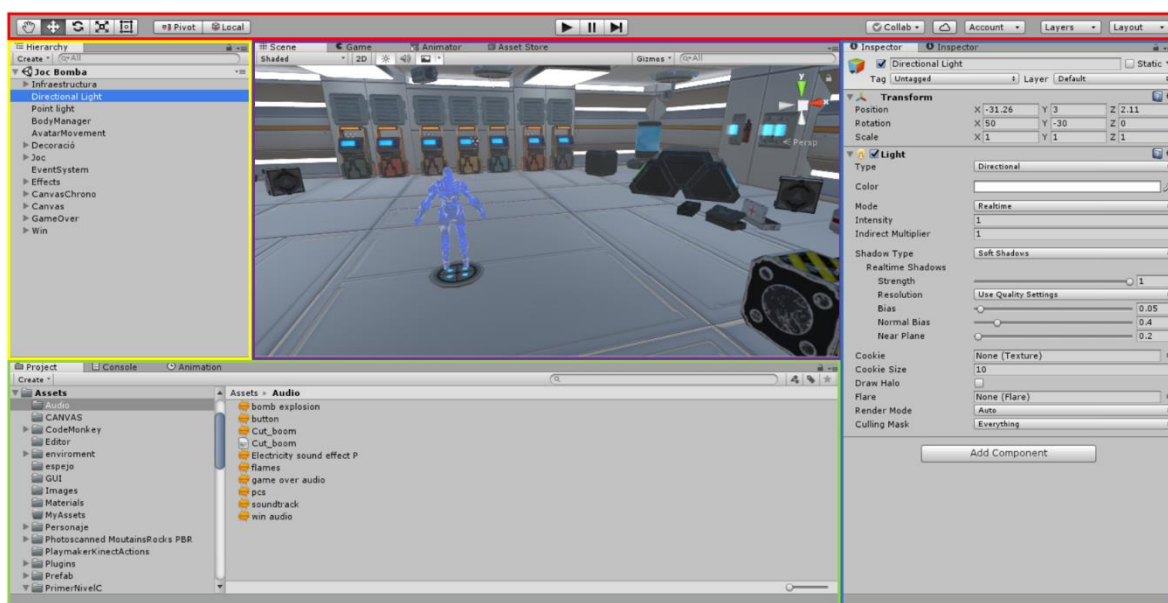


Figura 4.2 Interfície de Unity

## 4.2 Llenguatges de programació

**C#** és un llenguatge de programació orientat a objectes, dissenyat per la distingida companyia d'informàtica Microsoft [19].

La programació orientada a objectes és una branca de la informàtica que utilitza els objectes i les interaccions d'aquests per poder dissenyar aplicacions i programes informàtics. Cal recordar que en programació, un objecte és una entitat que reuneix l'estat (que són les dades de l'objecte), el comportament o mètode (són les que defineixen les operacions que pot fer l'objecte en qüestió) i la identitat (és el component diferenciador dels altres objectes).

**PHP**, que és l'acrònim de "Hypertext Preprocessor", és un llenguatge de codi obert especialment adequat pel desenvolupament de pàgines web [20].

Un dels avantatges que ofereix la programació en PHP és que en comptes d'utilitzar varies comandes per mostrar el HTML, com succeeix, per exemple, amb el llenguatge en C, aquest ja conté el HTML amb codi incrustat, de manera que permet reduir les línies de codi.

**SQL**, que és l'acrònim de "Structured Query Language", és un llenguatge estàndard i interactiu que dona accés a bases de dades relacionals i que permet especificar tot un variat d'operacions en ella [21].

A més, és un llenguatge que brinda la possibilitat de tenir accés i manipulació de dades a una base de dades, a part de que també es pot integrar a altres llenguatges de programació i combinar amb qualsevol base de dades específica, com és en el cas del MySQL.

### 4.3 XAMPP i phpMyAdmin

XAMPP és un paquet de programari lliure, que conté el servidor HTTP de codi obert multiplataforma Apache, el sistema de gestió de base de dades MySQL i les eines necessàries per poder utilitzar els llenguatges de programació PHP i Perl. D'aquesta manera, és possible crear una base de dades local MySQL amb el XAMPP i gestionar-la amb el servidor HTTP Apache.

Així doncs, el primer que s'ha de fer és activar els mòduls de l'Apache i MySQL des del panell de control del XAMPP (a la columna "Actions", requadre blau de la Figura 4.3, clicar a l'"Start" dels mòduls corresponents). A continuació, s'habilita el mòdul MySQL, clicant a l'enllaç "Admin" (requadre vermell de la Figura 4.3), que enllaça amb la pàgina web phpMyAdmin, que és on es permet crear, modificar i eliminar les bases de dades.

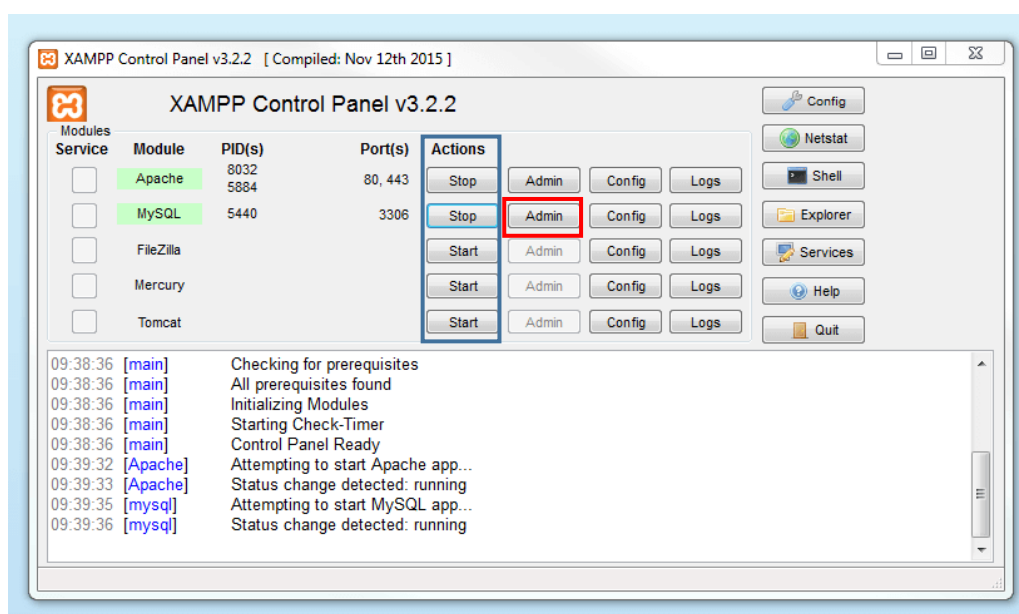


Figura 4.3 Pantalla d'inici XAMPP

phpMyAdmin és una eina de software escrita amb llenguatge PHP (software de codi obert), que gestiona l'administració de MySQL a través de diverses pàgines web. La seva pàgina principal està formada per varis menús i assistents que fan que la interacció amb el software sigui més fàcil i intuïtiva per l'usuari i que, evidentment, cal esmenar [22]. La barra lateral esquerra (requadre de color vermell



de la Figura 4.4); ofereix certes dreceres, com anar a la pàgina d'inici, canviar d'usuari, accedir el manual del programa phpMyAdmin, entre d'altres. També mostra les bases de dades a les que es tenen accés amb l'usuari creat.

Just a la part superior del menú principal (requadre de color blau de la Figura 4.4) es situa la barra del menú d'opcions, on es pot accedir als diferents assistents i a la informació d'estat del servidor de bases de dades. I per últim, es troba l'àrea de detall i de treball (requadre de color verd de la Figura 4.4).

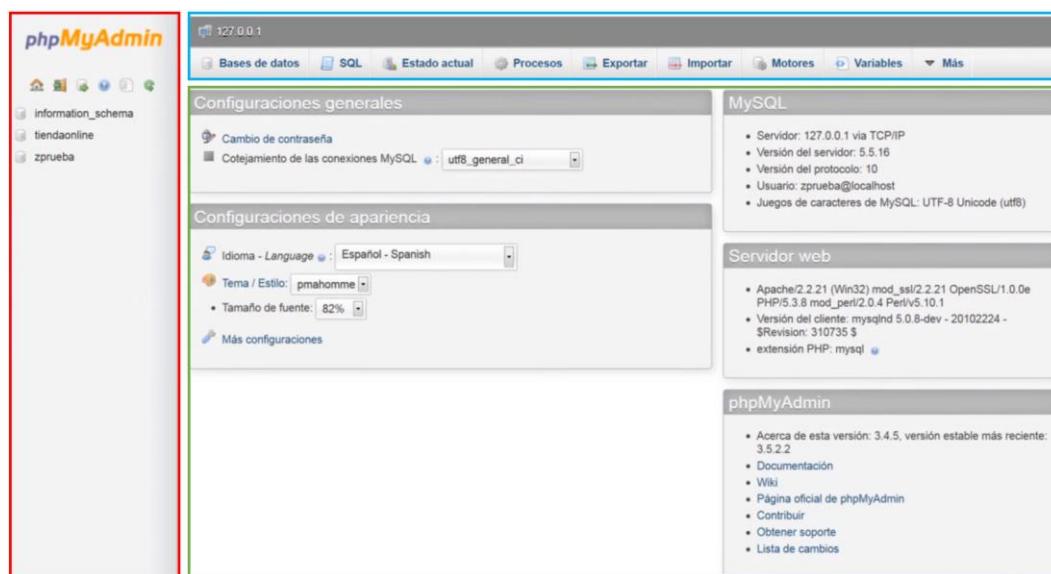


Figura 4.4 Menú principal phpMyAdmin

Cada base de dades està formada per un conjunt de taules [23]. Per crear una nova taula a una base de dades, primerament s'ha d'escollir el nom i el nombre de columnes que té (Figura 4.5).

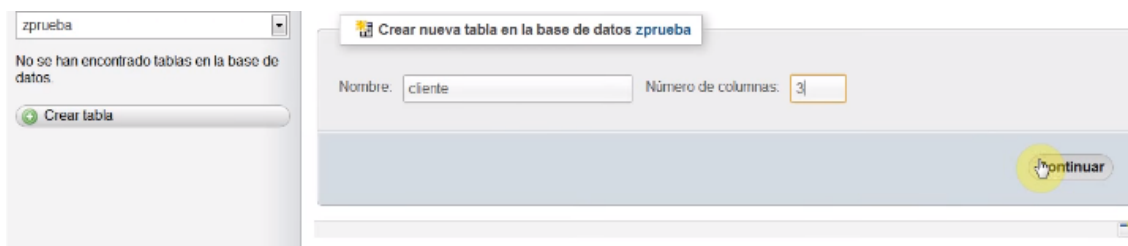


Figura 4.5 Creació nova taula

A continuació, l'usuari podrà afegir totes les files que necessiti com camps requereixi. Per a cada fila que insereixi, haurà de definir el tipus de dada de la columna i la longitud màxima de les dades. Les dades determinen com s'emmagatzemen físicament les peces d'informació de la base de dades que s'està creant, com es comparen i ordenen i com es presenten a la sortida. El tipus de dades més usuals són els números (*bit*, *bool*, *int*, *float*, ...), les dates (*date*, *time*, *datetime*, ...) i les cadenes de caràcters (*text*, *varchar*, *set*, *enum*, ...).

La columna A\_I, que es pot veure a la Figura 4.6, serveix per a totes aquelles dades on el valor s'incrementa de manera automàtica, com podria ser l'identificador (ID) del pacient o del fisioterapeuta.

L'usuari també haurà d'especificar el camp que tindrà el valor de la clau primària, que n'hi poden haver varies per taula. La clau primària és el camp que identifica de forma única un registre en una taula (en el treball en qüestió, un exemple seria l'ID d'un fisioterapeuta, que és propi per a cadascun d'ells). D'aquesta manera, no hi haurà cap altre registre que pugui tenir la mateixa clau primària.

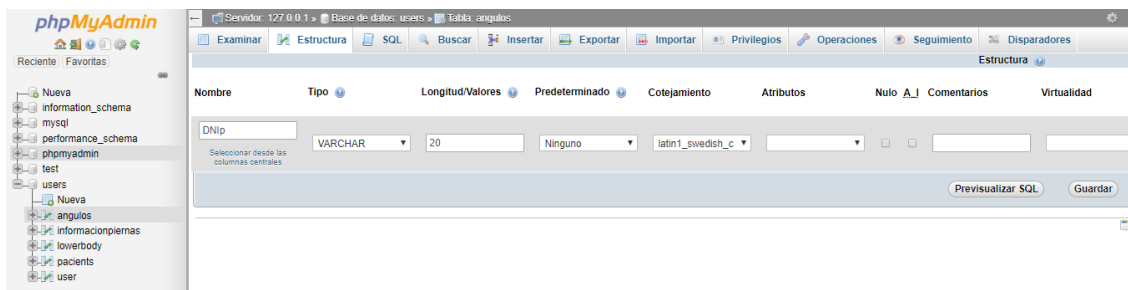


Figura 4.6 Creació nova columna

Aquestes claus primàries són de gran importància per poder relacionar les taules entre sí. Per poder completar aquestes relacions es requereixen de claus alienes. La clau aliena identifica un camp que pertany a una altra taula, el qual el camp en particular ha de ser clau primària en aquella taula. De forma resumida, les claus alienes s'utilitzen per fer una connexió amb claus primàries que pertanyen a altres taules. D'aquesta manera, és possible optimitzar i facilitar la consulta de dades i evitar la duplicitat de registres.

## 4.4 Kinect

El sensor de moviment Kinect és un dispositiu creat per l'empresa Microsoft, destinat per a la seva videoconsola Xbox 360, el qual es va sortir al mercat per primera vegada el 4 de novembre del 2010 als Estats Units d'Amèrica. La intenció de Microsoft era oferir al seu públic una experiència innovadora i totalment diferent a qualsevol altra consola. La seva tecnologia permetia el reconeixement corporal en 3D, així com també un reconeixement facial i de veu, tot a partir d'una càmera RGBD (capta l'ambient que rodeja a l'usuari) i d'una altra d'infrarojos, uns sensors de profunditat, un micròfon i un processador personalitzat [24].

Dos anys més tard, Microsoft ofereix el "Kinect for Windows", el qual autoritza el seu ús amb Windows PC. Tot i vendre prop de 35 milions d'unitats en 7 anys i batre un Rècord Guinness el 2011 com el dispositiu de consum de més ràpida venda (8 milions de Kinect venudes entre el 4 de novembre de 2010 i el 3 de gener de 2011) [25], la realitat és que no va acabar de quallar entre la comunitat de jugadors i el 2017 va donar lloc a la seva desaparició.



Figura 4.7. Kinect bat un Rècord Guinness el 2011 [26]

#### 4.4.1 Requisits per poder utilitzar el Kinect

A la taula (Taula 4.1) que es mostra a continuació indica els requisits mínims que ha de tenir l'ordinador per a la correcta utilització del Kinect [27].

Requisits del sistema operatiu	<ul style="list-style-type: none"> <li>- Windows 8 (64 bits)</li> <li>- Windows 8.1 (64 bits)</li> <li>- Windows Embedded Standard 8 (64 bits)</li> <li>- Windows Embedded Standard 8.1 (64 bits)</li> </ul>
Requisits del hardware	<ul style="list-style-type: none"> <li>- Processador de 64 bits (x64)</li> <li>- Processador de doble nucli a 3,2 GHz o superior</li> <li>- Bus USB 3.0</li> <li>- 2 GB de memòria RAM</li> </ul>
Requisits de software	<ul style="list-style-type: none"> <li>- Microsoft Visual Studio 2013 Express o una versió més recent</li> <li>- Microsoft Visual Studio 2012 Express o una versió més recent</li> <li>- .NET Framework 4.0</li> <li>- Microsoft Speech Platform SDK v11</li> </ul>

Taula 4.1 Requisits Kinect

#### 4.4.2 Especificacions tècniques

En aquest apartat es mostra els components del qual està format el sensor Kinect, així com les seves especificacions tècniques.

Els components més rellevants són els següents[28]:

- Càmera RGB: permet obtenir imatges amb colors. La imatge està formada per un conjunt de píxels, el qual cada un d'ells està compost per quatre components, que representa els valors dels colors bàsics (vermell, verd, blau) i la representació de la transparència.
- Emissors d'infrarojos: és l'encarregat de reconèixer cada píxel, el qual cadascun d'ells és agregat a un vector de píxels (cada píxel té un pes de 2 bytes) que conforma la imatge final. Aquesta imatge conforma 2 bytes extra, que és on s'emmagatzema la informació de la distància entre el píxel i el sensor de profunditat.
- Sensor de profunditat: permet mesurar la distància entre la persona o l'objecte que reconeix i el propi Kinect. La manera pel qual calcula aquesta distància és a partir del raig infraroig, el qual l'emissor i el receptor es coordinen de tal manera que, com el receptor coneix quan tarda en rebre la senyal de tornada, es sap a on ha rebotat la senyal i, per consegüent, sap la profunditat.
- Punts de reconeixement: és el sensor que permet reconèixer els moviments d'una persona a partir de la detecció de punts. D'aquesta manera, amb la unió dels punts és com es forma el cos d'una persona i representa el seu moviment.

Tot seguit, es presenta una taula (Taula 4.2) amb les característiques tècniques del Kinect [29]:

<b>Càmera RGB</b>	1920 x 1080 píxels, a 30 <i>frames</i> per segon
<b>Imatge de profunditat</b>	512 x 424 píxels, a 30 <i>frames</i> per segon
<b>Màxima profunditat de distància</b>	Aproximadament 4.5 m
<b>Mínima profunditat de distància</b>	50 cm
<b>Inclinació motor</b>	No
<b>Número de punts definits en l'esquelet</b>	25
<b>Número de persones que detecta</b>	6
<b>U.S.B Standard</b>	3.0

Taula 4.2 Característiques tècniques Kinect

En la següent figura (Figura 4.8) es pot apreciar els 25 punts que detecta el sensor del cos humà i, com amb ells, crea l'esquelet.

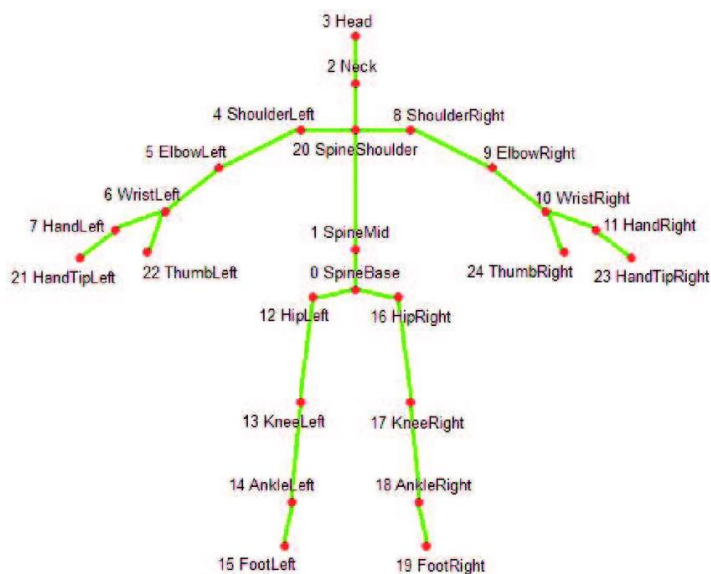


Figura 4.8 Punts detectats per Kinect [30]

### 4.4.3 Kinect per a Windows SDK 2.0

El SDK (*Software Development Kit*) és un conjunt d'eines i llibreries que ajuden al usuari a interactuar amb el sensor de moviment Kinect i, d'aquesta manera, brinda la possibilitat de poder desenvolupar una aplicació, com és en el cas d'aquest projecte [31].

Una vegada el Kinect ha detectat la persona o l'objecte que es troba davant seu, es facilita la informació detallada de la posició exacta a l'espai (punts X, Y i Z) de tots els punts que conforma el cos humà (els 25 punts que és capaç de detectar el sensor). La Figura 4.9 és la visió que té el Kinect al detectar una persona (sensor de profunditat). Com es pot veure, localitza els 25 punts que conforma el cos humà i els uneix en línies per acabar de donar forma al subjecte.

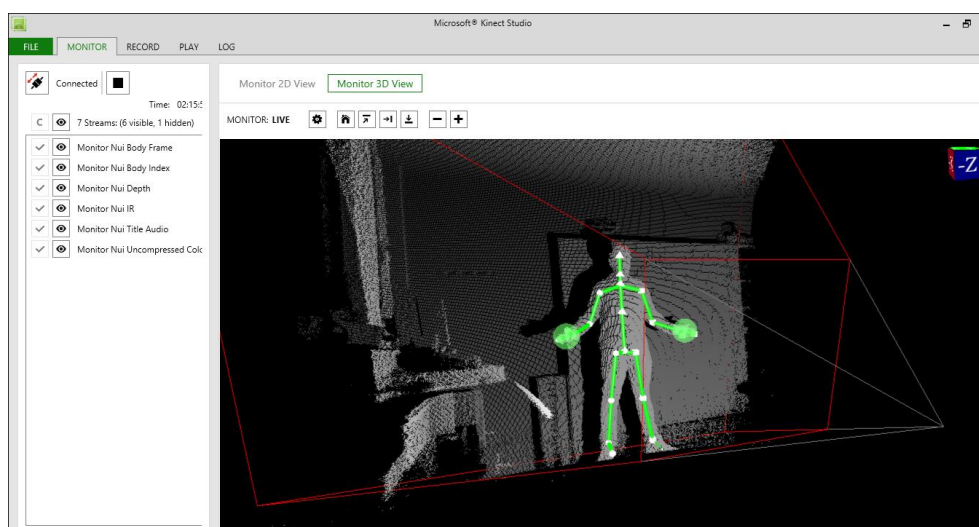


Figura 4.9 Pantalla Microsoft Kinect Studio

## 4.5 Ulleres HTC Vive

Les HTC Vive són unes ulleres de realitat virtual fabricades per la multinacional taiwanesa HTC i l'empresa de videojocs Valve. Com la gran majoria d'ulleres de RV, aquestes estan dissenyades per utilitzar l'espai en un indret tancat i submergir-se en un entorn virtual el qual permet a l'usuari caminar i utilitzar una sèrie de controladors per poder interactuar amb objectes virtuals [32].



Figura 4.10 Equipament de RV HTC Vive [33]

### 4.5.1 Requisits per poder utilitzar les HTC Vive

La Taula 4.3 mostra els requisits que hauria de tenir l'ordinador per a poder utilitzar les ulleres de RV HTC Vive:

<b>Targeta gràfica</b>	NVIDIA GeForce GTX 1060 o AMD Radeon RX 480. També són possibles targetes gràfiques equivalents o millors
<b>Processador</b>	Intel Core i5-4590 o AMD FX 8350, o un processador equivalent o millor
<b>Memòria</b>	4 GB de RAM
<b>Sortida de vídeo</b>	HDMI 1.4, DisplayPort 1.2 o més actualitzada
<b>Entrades USB</b>	1 x USB 2.0 o millor
<b>Sistema operatiu</b>	Windows 7 SP1, Windows 8.1, Windows 10

Taula 4.3 Requisits HTC Vive

### 4.5.2 Hardware de les HTC Vive

HTC afegeix al seu kit Vive un conjunt de hardwares per tal d'oferir una millor immersió en l'entorn de la RV. Aquests són els següents:

- Ulleres de RV: estan formades per un parell de pantalles OLED de 3.6 polsades, amb una resolució de 2160 x 1200 píxels (1080 x 1200 píxels per ull) i presenta un angle de visió de fins a 110 graus.
- Controladors: es tracta de dos comandaments sense fil, format per un conjunt de sensors de moviment i de botons, que serveixen per poder controlar les accions i els moviments dins de la RV sense la necessitat de tenir un teclat o ratolí.
- Estacions base: estan formades per dos sensors d'infrarojos que tenen la finalitat de detectar la posició de l'usuari. Així doncs, s'encarreguen d'enviar els raigs infrarojos a les ulleres i als controladors per tal de reproduir de forma exacta qualsevol moviment del subjecte a l'aplicació o al joc. A més, una vegada es registra la posició d'aquestes respecte el de les ulleres, s'envia aquestes coordenades a l'ordinador i, d'aquesta manera, es poden fixar les posicions de les ulleres i de les estacions per poder crear un espai virtual.

#### 4.5.3 Software necessari pel funcionament del hardware

El kit complet de HTC Vive necessita d'un software necessari per a poder controlar tot el sistema de processament i visió; es tracta del Steam VR, desenvolupat per Valve Corporation i llançat al mercat per l'empresa Steam el 2014 [34]. S'encarrega de registrar tots els moviments de l'usuari, mitjançant les estacions base, i de seguir la seva posició al llarg de tota l'habitació de manera exacta. Així doncs, si l'usuari s'apropés a una paret, aquest no toparia ja que a través de les ulleres de RV també veuria que s'està aproximant a la paret en qüestió (la càmera de l'aplicació, que permet veure el que s'està reproduint en les ulleres de RV, és capaç de seguir en tot moment al jugador).

Una vegada s'engega el programa i tots els components de l'HTC Vive estan connectats correctament a l'ordinador, apareix una pestanya com el de la Figura 4.11Figura 4.11, indicant que tot el hardware està preparat pel seu funcionament.

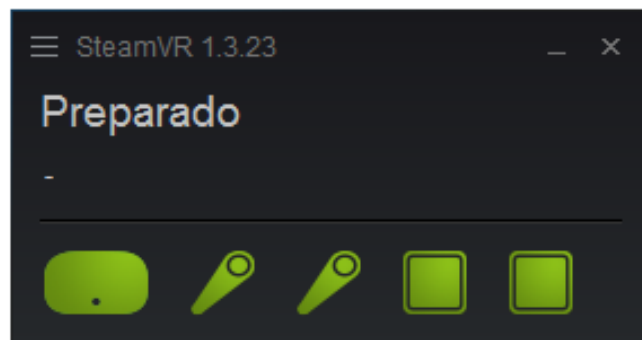


Figura 4.11. Pestanya informativa del Steam VR

Però en el cas que algun dels components de l'HTC Vive no s'hagués connectat, l'usuari pot saber perfectament quin o quins hardwares no estan connectats perquè la seva silueta no sortiria il·luminada

(en la Figura 4.12, l'estació base és la que no està connectada a l'ordinador). A més, també indica la bateria que li resta a cada controlador, com es pot observar a la Figura 4.13.

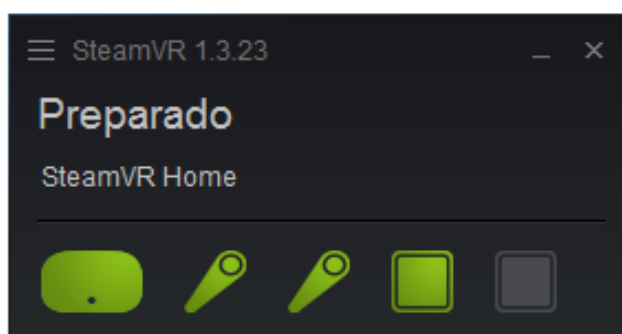


Figura 4.12. Indicació de no connectat d'una de les estacions base

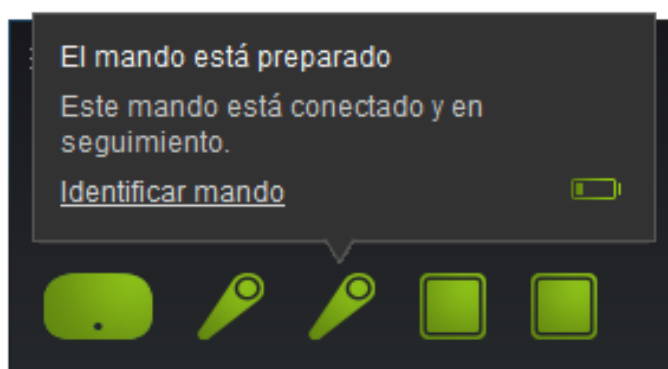


Figura 4.13. Indicació de bateria d'un dels controladors



## 5. Estructura i funcionalitats de l'aplicació

Aquest apartat té com a finalitat aclarir l'estructura i les funcionalitats de l'aplicació, a la Figura 5.1 es mostra un diagrama de blocs resumint l'aplicació:

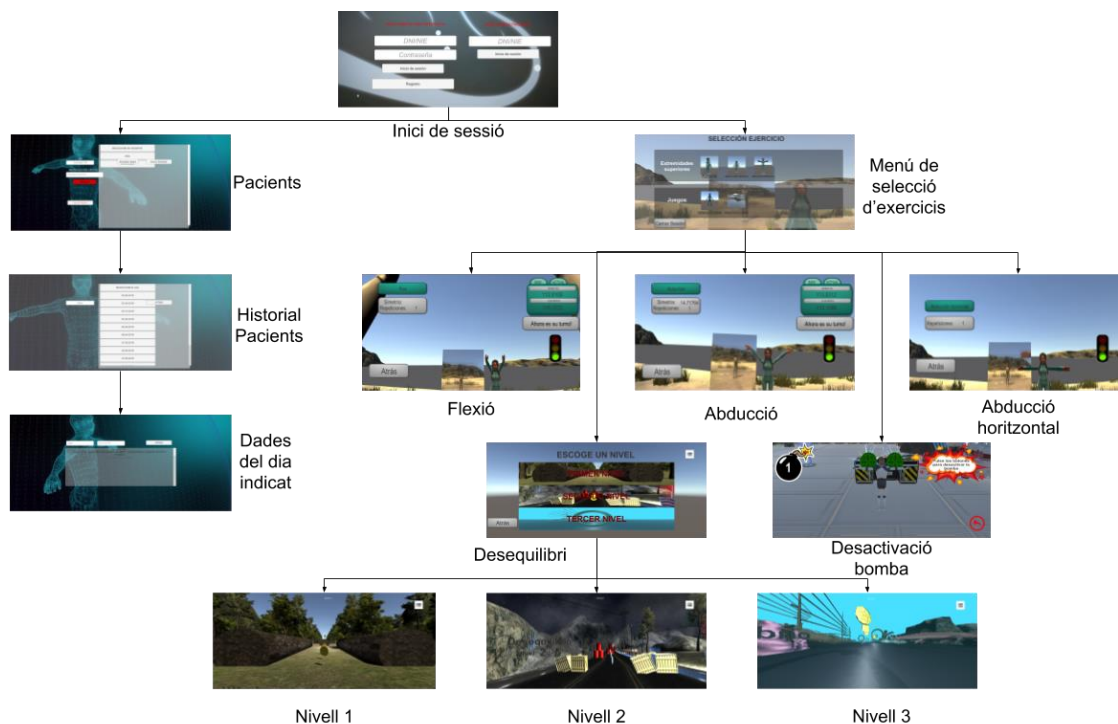


Figura 5.1 Estructura aplicació

A la part superior s'aprecia el menú principal i, a partir d'aquí, hi ha una bifurcació. En primer lloc, a la part esquerra, es mostra la zona per als fisioterapeutes. A l'introduir el DNI i la contrasenya ja definida del fisioterapeuta en qüestió s'accedeix a una pantalla amb els seus pacients. Al clicar a algun dels pacients apareix una altra finestra on es pot veure l'historial d'aquests, amb els dies que han fet el tractament. A continuació, al clicar un dels dies apareixen les dades específiques de les sessions (l'hora i l'angle màxim per a l'exercici de flexió, tan per al braç dret com per l'esquerra).

A l'entrar a l'aplicació amb el DNI del pacient es condueix a un menú de selecció d'exercicis i jocs, on es poden trobar tres exercicis (flexió, abducció i abducció horitzontal) i dos jocs. Un corresponen en exercicis relacionats amb l'equilibri corporal amb els seus respectius tres nivells i l'altre consisteix en treballar els exercicis de flexió dels braços, l'entorn lúdic es basa en la desactivació de la bomba.

Per la part que pertoca a aquest projecte s'ha treballat en la creació de l'exercici d'abducció horitzontal i el joc de desactivació de bomba. La resta de l'aplicació estava feta per antics alumnes o en el cas del joc de l'equilibri corporal ha estat creat per els alumnes de l'altre grup d'aquest quadrimestre.

A continuació es procedeix a explicar en primer lloc l'exercici d'abducció horitzontal:

## 5.1 Abducció horitzontal

Aquest exercici té com a objectiu la realització del moviment que es pot veure a la Figura 5.2.

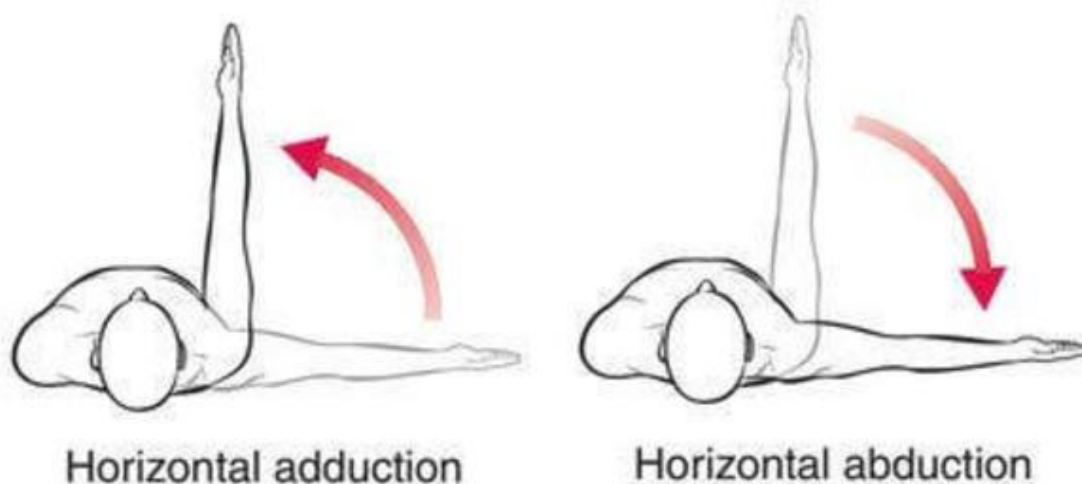


Figura 5.2 Exercici d'abducció horitzontal [35]

En primer lloc es comença des d'un estat de repòs. En el moment que el semàfor de la pantalla es posa de color verd, l'anomenada fisioterapeuta virtual comença a realitzar el moviment. És en aquest moment quan l'usuari ha d'intentar realitzar aquest moviment amb la màxima similitud possible respecte a la monitora virtual.

A la Figura 5.3 es pot observar la pantalla de l'exercici, també anomenat *canvas*, on es poden apreciar les següents seccions:

- El nom de l'exercici, situat al requadre turquesa
- El comptador de repeticions, que es troba just a sota del requadre del nom de l'exercici
- El semàfor, que indica l'inici de l'exercici
- I, per últim, el botó "Atrás", que és de vital importància, ja que a part de la finalitat òbvia, que és tornar al menú anterior, és activar els programes indicats per enviar les dades a la base de dades (s'envia el nombre de repeticions i el temps de realització).



*Figura 5.3 Canvas abducció horitzontal*

## 5.2 Flexió braços (desactivació bomba)

La Figura 5.4 mostra l'estructura i funcionament d'aquest joc:

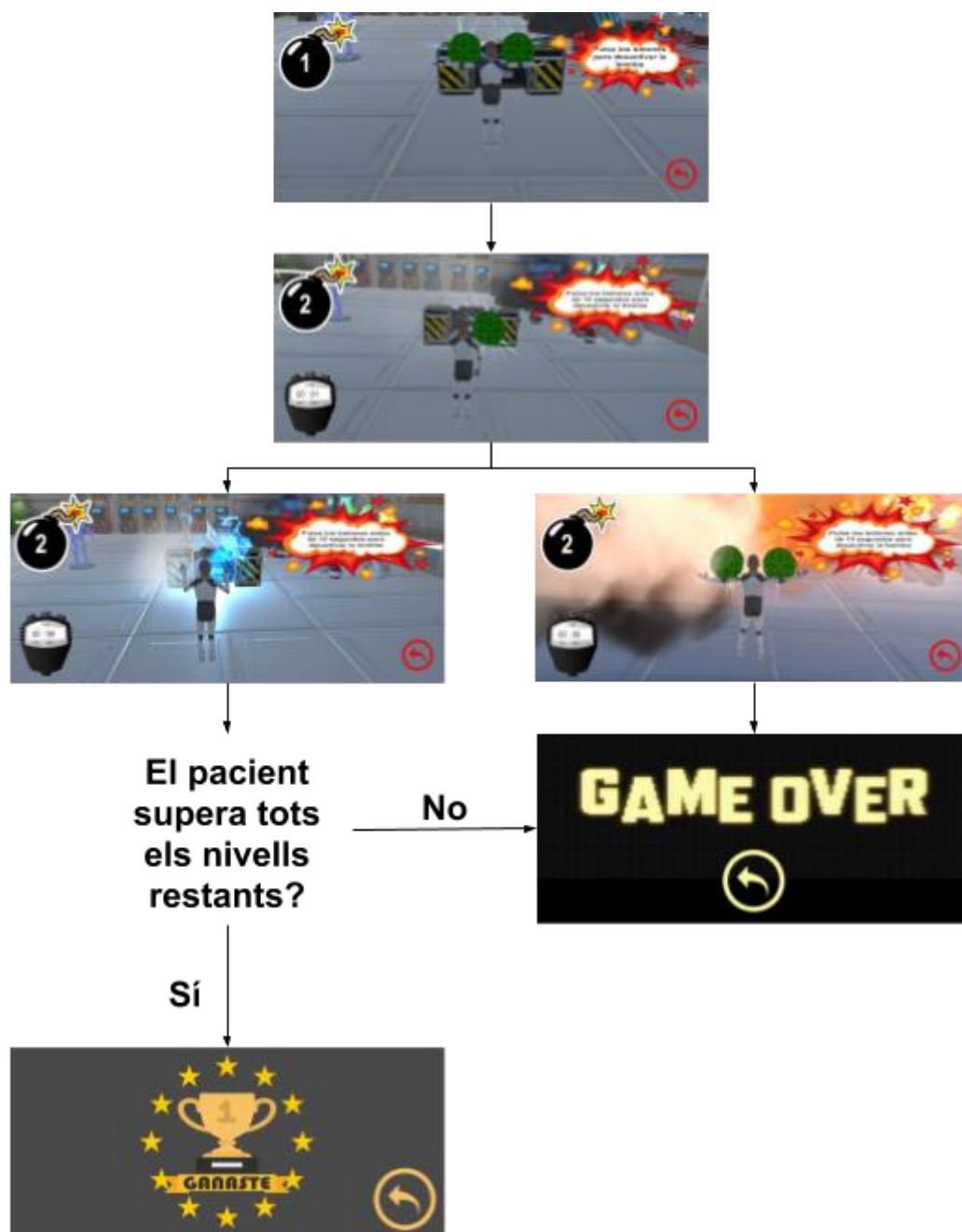


Figura 5.4 Esquema desactivació bomba

La primera imatge del diagrama mostra el nivell 1, el qual és un nivell introductori per tal que l'usuari entri en la dinàmica del joc. A la part superior esquerra d'aquesta imatge s'aprecia el dibuix d'una

bomba amb un número, que és l'indicador de nivell. A la part superior dreta, es mostra el dibuix d'una explosió, el qual aporta al participant les instruccions per a superar el nivell i, finalment, a la part inferior dreta, un últim botó que representa novament la tornada al menú de selecció i l'enviament de dades a la base de dades (en el cas d'aquest joc, s'envia el nivell en el que s'ha clicat el botó i el temps de realització).

El joc és molt senzill, es tracta de tocar dos pulsadors de color verd que es mantenen al voltant del personatge virtual del pacient. Al realitzar el contacte amb aquests, representa que la bomba es desactiva i s'accedeix al següent nivell.

Ja que el joc explicat fins ara no hi ha manera de perdre, s'ha fet el següent: els nivells amb números senars seran introductoris, és a dir, no es pot perdre. D'aquesta manera, el pacient es podrà fer una idea de l'alçada dels pulsadors sense cap pressió. Un cop superat aquest nivell, es repetirà el mateix que abans però amb un petit estímul, un compte enrere de 15 segons. Si abans de que aquest compte enrere acabi es pitgen els 2 botons, la bomba es desactiva; en canvi, si en 15 segons no es supera el nivell, la bomba explotarà i apareixerà la pantalla de "Game Over", la qual només deixa retrocedir per tornar al menú.

El joc consta de 8 nivells, és a dir, 4 alçades diferents. En els 2 primers nivells, l'alçada és igual a l'alçada del punt central del cap; en el nivell 3 i 4 l'alçada passa a ser un 20% de l'original; els nivells 5 i 6 s'afegeix un 30% a l'alçada original i, per a finalitzar, els nivells 7 i 8 tenen un alçada superior en un 40% a la del cap.

### 5.3 Càlcul d'angles

S'ha calculat dos angles, ja que d'aquesta manera es pot definir l'orientació d'un segment. El primer aporta informació sobre l'angle d'abducció (Figura 5.2) i l'altre sobre l'angle de flexió (Figura 5.5).

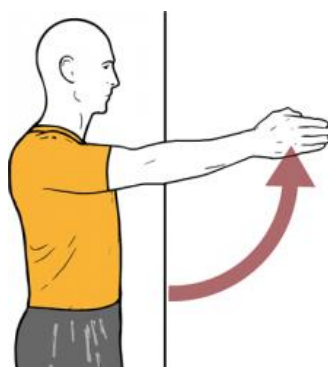


Figura 5.5 Angle de flexió [36]

Per poder arribar a calcular aquests angles s'ha treballat amb el càlcul d'angles entre vectors, el qual es defineix per la següent fórmula:

$$\theta = \cos^{-1} \left( \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|} \right) \quad \{on \theta \text{ és l'angle i } \vec{A} \text{ i } \vec{B} \text{ els vectors en qüestió}\}$$

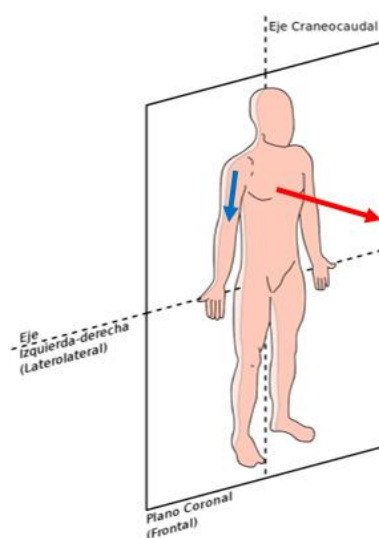
*Fórmula 5.1 Càlcul de l'angle entre 2 vectors*

Tot i això, no s'ha utilitzat aquesta fórmula matemàtica literalment, ja que Unity3D té una funció anomenada "Vector3.Angle" que implementa l'anterior fórmula, la qual a partir de 2 vectors de 3 components retorna l'angle més petit entre ells (és a dir, sempre menor a 180°). Per aquest motiu s'ha obtingut 2 vectors per a cadascun dels 2 angles i així fer possible el seu càlcul.

### 5.3.1 Angle d'abducció

Per al càlcul de l'angle d'abducció s'han creat 2 vectors: un que avança des del punt de l'espatlla fins al colze i un altre que representa el vector normal al pla frontal (ambdós representats en color blau i vermell respectivament a la Figura 5.6). Per a poder representar aquest vector, s'ha hagut de crear aquest pla mitjançant 3 punts del cos, els 2 de les espatlles i l'anomenat punt "SpineMid" (es poden veure representats a la Figura 4.8). Tot aquest procés es realitza de forma senzilla gràcies a la funció "Plane.Set3Points" de Unity3D, que a partir de 3 punts crea un pla. Unity3D té una funció que també és de gran utilitat, es tracta de la funció "Plane.normal", que aporta un vector normal a un pla determinat.

Un cop determinats aquests 2 vectors, s'ha utilitzat la funció "Vector3.Angle", esmentada anteriorment, per tal d'obtenir aquest angle d'abducció.



*Figura 5.6 Representació pla frontal [37]*

### 5.3.2 Angle de flexió

Com en l'angle d'abducció, s'ha creat un vector entre espatlla i colze per a tots 2 braços (vector blau a la Figura 5.7). En aquest cas, el que canvia és l'altre vector; d'aquesta manera, s'ha generat un vector perpendicular al pla sagital (vector vermell a la Figura 5.7), tot i que en aquest cas no ha sigut necessària la creació d'aquest pla, ja que s'ha utilitzat directament el vector entre les 2 espatlles.

Així doncs, s'ha utilitzat altra vegada la funció "Vector3.Angle", entre el vector normal al pla sagital i el vector d'espatlla a colze.

És important destacar que, al contrari de l'angle d'abducció, en aquí el sentit del vector normal al pla pot portar nombrosos problemes, ja que pot tenir el mateix sentit que un braç aleshores el contrari per a l'altre. És per això que, per al càlcul de l'angle de flexió en el braç dret, s'ha utilitzat el vector que va d'espatlla dreta a l'esquerra i, per al braç esquerre, tot el contrari (vector d'espatlla esquerra a dreta).

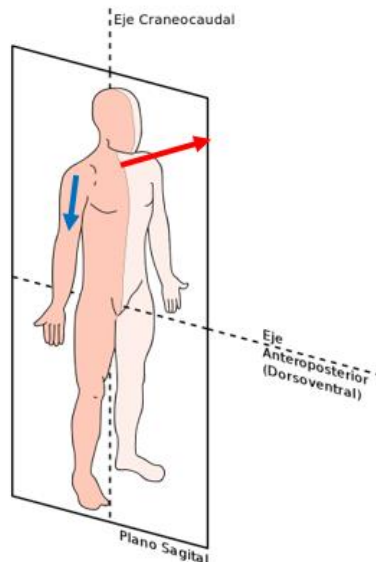


Figura 5.7 Representació pla sagital [37]

## 6. Metodologia del disseny de l'aplicació

### 6.1 Abducció horitzontal

La Figura 6.1 mostra el menú de selecció d'exercicis i jocs de l'aplicació creada. La Figura 6.2 mostra les accions que comporta el fet de clicar el botó de selecció de l'exercici d'abducció horitzontal, la penúltima acció d'aquest botó comporta l'activació del programa "HABDUPP", que es situa dins del codi "BodySourceUpperBody" (que és el principal codi utilitzat durant el transcurs del projecte). A més a més, es desactiva el *Game Object* del *canvas* "Selección Movimiento". Per tant, desapareixerà el panell que hi havia anteriorment i s'activarà el *canvas* de l'exercici en qüestió (Figura 6.3), que és el que mostra el nom de l'activitat, el número de repeticions, el semàfor i el botó per a tornar al menú de selecció de moviments.



Figura 6.1. Menú de selecció d'exercicis

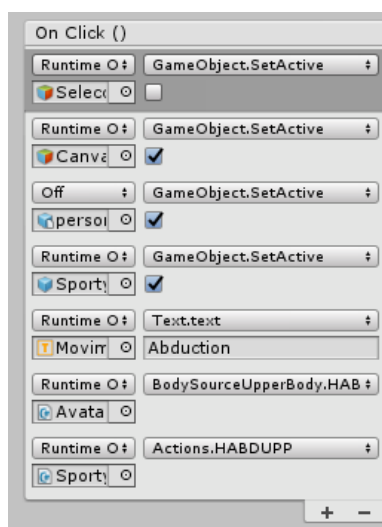


Figura 6.2 Accions al clicar el botó d'abducció horitzontal





Figura 6.3 Canvas abducció horitzontal

La Figura 6.4 presenta la funció “HABDUPP” dins del codi “BodySourceUpperBody”, activada anteriorment al clicar sobre el botó de l'exercici d'abducció horitzontal, encarregada d'activar la variable “HABDUP”, de començar la corutina “SeemaforoHABDUP” i de donar-li al cronòmetre un valor inicial de 0 segons.

```
public void HABDUPP()
{
    HABDUP = true;
    StartCoroutine(SeemaforoHABDUP());
    HABDUPtime = 0;
}
```

Figura 6.4 Programa “HABDUPP”

La corutina “seemaforoHABDUP” (Figura 6.5) funciona de la següent manera: en primer lloc espera tres segons i, a continuació, desactivar el *Game Object* “semaforoHABDUP” (que és la imatge del semàfor en vermell), per posteriorment activar un altre *Game Object* anomenat “semaforo2HABDUP” (que és la imatge del semàfor en verd).

```
IEnumerator SeemaforoHABDUP()
{
    yield return new WaitForSeconds(3);

    semaforoHABDUP.SetActive(false);
    semaforo2HABDUP.SetActive(true);
}
```

Figura 6.5 Corutina "SeemaforoHABDUP"

Dins la funció "Update", que igual que les anteriors funcions i corutines es troba dins del codi "BodySourceUpperBody", s'utilitza la variable "HABDUptime" (Figura 6.6), aquesta funció "Update" duu a terme totes les línies de codi que es situen dins seu en cada *frame*, és a dir s'estarà executant *continuament*; per aquesta raó s'afegeix aquesta línia de codi, que suma el temps de cada *frame* en aquell instant. El que no es pot fer és sumar-li sempre el mateix valor, ja que la longitud temporal de cada *frame* no acostuma a ser la mateixa.

```
void Update()
{
    HABDUptime += Time.deltaTime;
}
```

Figura 6.6 "HABDUptime" dins d'"Update"

A més a més, dins d'aquesta funció "Update" hi ha la condició "if HABDUP == true" que es mostra en la Figura 6.7 (la principal de l'exercici d'abducció horitzontal).

Primer es declaren les variables per a les tres coordenades per a cadascun dels punts que dona el sensor de moviment Kinect (es pot veure al requadre verd de la Figura 6.7). Llavors, es declara un vector de tres valors amb les coordenades x, y i z de cadascun dels punts (s'observa al requadre vermell de la Figura 6.7). A continuació, es creen tres vectors (requadre violeta de la Figura 6.7): un entre els punts de les espatlles (de dreta a esquerra), un altre que uneix els punts entre l'espatlla dreta al colze dret i un últim que va des de l'espatlla esquerra al colze esquerre, anomenats "shoulders", "rightarm" i "leftarm" respectivament.

A continuació (requadre groc de la Figura 6.7), es crea un nou pla anomenat "Chest" a partir de les dues espatlles i el punt central de la columna. Es crea un vector perpendicular a aquest pla anomenat "ChestVector", usant la funció "Chest.normal". Llavors es calcula l'angle d'abducció (entre aquest vector "ChestVector" i cadascun dels vectors ja definits anteriorment com a "leftarm" i "rightarm") amb la funció (requadre blau de la Figura 6.7).

Per altra banda, per calcular l'angle de flexió es va realitzar mitjançant el vector de les espatlles i el vector de cadascun dels braços (destacar que el vector "shoulders" s'ha d'invertir per al braç esquerra per a obtenir un angle coherent amb el del braç dret).

```

if (HABDUP == true)
{
    float x1 = body.Joints[JointType.ShoulderRight].Position.X;
    float y1 = body.Joints[JointType.ShoulderRight].Position.Y;
    float z1 = body.Joints[JointType.ShoulderRight].Position.Z;

    float x2 = body.Joints[JointType.ShoulderLeft].Position.X;
    float y2 = body.Joints[JointType.ShoulderLeft].Position.Y;
    float z2 = body.Joints[JointType.ShoulderLeft].Position.Z;

    float a1 = body.Joints[JointType.WristRight].Position.X;
    float b1 = body.Joints[JointType.WristRight].Position.Y;
    float c1 = body.Joints[JointType.WristRight].Position.Z;

    float a2 = body.Joints[JointType.WristLeft].Position.X;
    float b2 = body.Joints[JointType.WristLeft].Position.Y;
    float c2 = body.Joints[JointType.WristLeft].Position.Z;

    float a3 = body.Joints[JointType.SpineMid].Position.X;
    float b3 = body.Joints[JointType.SpineMid].Position.Y;
    float c3 = body.Joints[JointType.SpineMid].Position.Z;

    float a4 = body.Joints[JointType.ElbowRight].Position.X;
    float b4 = body.Joints[JointType.ElbowRight].Position.Y;
    float c4 = body.Joints[JointType.ElbowRight].Position.Z;

    float a5 = body.Joints[JointType.ElbowLeft].Position.X;
    float b5 = body.Joints[JointType.ElbowLeft].Position.Y;
    float c5 = body.Joints[JointType.ElbowLeft].Position.Z;

    float a6 = body.Joints[JointType.SpineBase].Position.X;
    float b6 = body.Joints[JointType.SpineBase].Position.Y;
    float c6 = body.Joints[JointType.SpineBase].Position.Z;

    float a7 = body.Joints[JointType.Neck].Position.X;
    float b7 = body.Joints[JointType.Neck].Position.Y;
    float c7 = body.Joints[JointType.Neck].Position.Z;

    float a8 = body.Joints[JointType.Head].Position.X;
    float b8 = body.Joints[JointType.Head].Position.Y;
    float c8 = body.Joints[JointType.Head].Position.Z;

    Vector3 shoulderright = new Vector3(x1, y1, z1);
    Vector3 shoulderleft = new Vector3(x2, y2, z2);
    Vector3 wristright = new Vector3(a1, b1, c1);
    Vector3 wristleft = new Vector3(a2, b2, c2);
    Vector3 spinemidv = new Vector3(a3, b3, c3);
    Vector3 Elbowright = new Vector3(a4, b4, c4);
    Vector3 Elbowleft = new Vector3(a5, b5, c5);
    Vector3 SpineB = new Vector3(a6, b6, c6);
    Vector3 neck = new Vector3(a7, b7, c7);
    Vector3 head = new Vector3(a8, b8, c8);

    Vector3 shoulders = new Vector3(x2 - x1, y2 - y1, z2 - z1);
    Vector3 rightarm = new Vector3(a4 - x1, b4 - y1, c4 - z1);
    Vector3 leftarm = new Vector3(a5 - x2, b5 - y2, c5 - z2);

    Plane Chest = new Plane();
    Chest.Set3Points(shoulderright, shoulderleft, spinemidv);
    Vector3 ChestVector = new Vector3();
    ChestVector = -Chest.normal;

    float HABDUPleft = Vector3.Angle(ChestVector, leftarm);
    float HABDUPright = Vector3.Angle(ChestVector, rightarm);
    float FLEXleft = Vector3.Angle(-shoulders, leftarm);
    float FLEXright = Vector3.Angle(shoulders, rightarm);

    if (Mathf.Abs(HABDUPright) < 30 && rep == true) Condició 1
    {
        if (repeticiones0 == true) Condició 1.1
        {
            conta = conta + 1;
        }
        rep = false;
    }

    if (Mathf.Abs(HABDUPright) > 30) Condició 2
    {
        rep = true;
        repeticiones0 = true;
    }

    var exp = repeticonesHABDUP.GetComponent<Text>();
    exp.text = (""+conta);
    repHABDUP = exp.text;

    StartCoroutine(TemporizadorHABDUP(HABDUPleft, HABDUPright));
}

```

Figura 6.7 Condició "if (HABDUP == true)"

Més endavant s'aprecia una altra condició (que s'anomenarà 1 per evitar possibles confusions), que s'utilitza per al càlcul de les repeticions. En primer lloc, la variable "rep", és una variable booleana i a l'inicialitzar el codi, es defineix com a vertadera. La primera condició (condició 1) estableix que si l'angle d'abducció del braç dret és inferior a 30° i "rep" és vertader (cal recordar que aquest angle assoleix els seus màxims en la obertura total dels braços), aleshores la variable "rep" serà falsa i, a més a més, existeix una altra condició (condició 1.1) dins d'aquesta que si "repeticones0" és vertader ("repeticones0" és una variable booleana que s'inicialitza com a falsa), el contador sumarà 1.

Per altra banda, es té una altra condició (que s'anomenarà condició 2) que estableix que si l'angle d'abducció del braç dret és superior a  $30^\circ$ , aleshores "rep" serà vertader i "repeticiones0" també ho serà.

És a dir, quan l'usuari comenci a fer l'exercici, s'activarà la condició 2 i, seguidament, es podrà accedir a la condició 1.1, ja que "repeticiones0" serà vertader. És en aquest punt on realment es fa la suma. D'aquesta manera es garantitza que la suma es farà en el tancament dels braços i que, si inicialment l'angle és inferior a  $30^\circ$ , no es compti com a repetició.

Un cop tancades les condicions anteriors, es pot apreciar la creació d'una variable anomenada "exp"; pren per valor la component de text del *Game Object* creat a l'inici del codi "repeticionesHABDUP". Aquest *Game Object* crea únicament un camp de text. A continuació, a la variable de text se li dona el valor de "conta", que és el comptador de repeticions i s'igual a el *Game Object* "repHABDUP" a la variable de text ("repHABDUP" és el que es mostra en el canvas de l'exercici en qüestió).

Per a finalitzar amb la condició de "HABDUP == true", s'envia totes les dades pertinents, és a dir, els quatre angles, tots els punts del cos declarats i el temps a la corutina "TemporizadorHABDUP".

Pel que fa a la corutina "TemporizadorHABDUP" (Figura 6.8), primer desactiva la variable "HABDUP", perquè durant el transcurs d'aquesta corutina no s'activi la condició anterior ("HABDUP == True"). A continuació, la corutina espera 0.2 segons i realitza una petita funció per a calcular si l'angle en aquell instant és el màxim o no. Si "HABDUPright" i "HABDUPleft" (que són els angles d'abducció en aquell instant) són majors a "a" i "a2" respectivament. Si són superiors al valor, aleshores "a" i/o "a2" prenen el valor de "HABDUPright" i/o "HABDUPleft", per tant el valor "a" i "a2" representen els valors màxims.

```
IEnumerator TemporizadorHABDUP(float HABDUPleft, float HABDUPright)
{
    HABDUP = false;
    yield return new WaitForSeconds(0.2f);

    if (HABDUPright > a)
    {
        a = HABDUPright;
    }

    if (HABDUPleft > a2)
    {
        a2 = HABDUPleft;
    }

    excelHABDUP(HABDUPright, HABDUPleft, FLEXright, FLEXleft,
    HABDUP = true;
```

Figura 6.8 Corutina "TemporizadorHABDUP"

Un cop realitzada aquesta operació s'envien els valors rebuts a una funció anomenada “excelHABDUP”, que envia els valors desitjats al directori d'un fitxer Excel.

Un cop dins la funció “excelHABDUP” (Figura 6.9), es crea la destinació del fitxer de Microsoft Excel, que en aquest cas és la drecera on es troben tots els arxius del projecte a la carpeta “Seguimiento”. Allà, dins d'aquesta, hi ha una carpeta amb el DNI del pacient en qüestió, i allà es troba finalment el fitxer amb el nom de l'exercici realitzat més la data de realització.

```
public void excelHABDUP(float HABDUPright, float HABDUPleft, float FLEXright, float FLEXleft, Vector3 s
{
    string path5 = Application.dataPath + "/Seguimiento/"+nombre+"/HAbduction" + dataDia + ".xls"; //Cr
    string content5 = HABDUptime+"\t"
        +shoulderright.x+"\t" + shoulderright.y + "\t" + shoulderright.z + "\t"
        + shoulderleft.x+ "\t"+shoulderleft.y + "\t" + shoulderleft.z+ "\t"
        + Elbowright.x + "\t" + Elbowright.y + "\t" + Elbowright.z + "\t"
        + Elbowleft.x + "\t" + Elbowleft.y + "\t" + Elbowleft.z + "\t"
        + wristright.x + "\t" + wristright.y + "\t" + wristright.z + "\t"
        + wristleft.x + "\t" + wristleft.y + "\t" + wristleft.z + "\t"
        + spinemidv.x + "\t" + spinemidv.y + "\t" + spinemidv.z + "\t"
        + SpineB.x + "\t" + SpineB.y + "\t" + SpineB.z + "\t"
        + neck.x + "\t" + neck.y + "\t" + neck.z + "\t"
        + head.x + "\t" + head.y + "\t" + head.z + "\t"
        + HABDUPright + "\t" + HABDUPleft + "\t"
        + FLEXright + "\t" + FLEXleft + "\n"; //Contenido del .xls

    if (!File.Exists(path5))
    {
        File.AppendAllText(path5, "Data Angulo Derecho\nLogin date: " + System.DateTime.Now + "\n" +
            "Tiempo \t"+
            "Hombro Derecho X \t Hombro Derecho Y \t Hombro Derecho Z \t" +
            "Hombro Izquierdo X \t Hombro Izquierdo Y \t Hombro Izquierdo Z \t" +
            "Codo Derecho X \t Codo Derecho Y \t Codo Derecho Z \t" +
            "Codo Izquierdo X \t Codo Izquierdo Y \t Codo Izquierdo Z \t" +
            "Muneca Derecha X \t Muneca Derecha Y \t Muneca Derecha Z\t" +
            "Muneca Izquierda X \t Muneca Izquierda Y \t Muneca Izquierda Z \t" +
            "Spine Mid X \t Spine Mid Y \t Spine Mid Z \t" +
            "Spine Base X \t Spine Base Y \t Spine Base Z \t" +
            "Cuello X \t Cuello Y \t Cuello Z \t" +
            "Cabeza X \t Cabeza Y \t Cabeza Z \t"+
            "Angulo HABDUP derecho \t Angulo HABDUP izquierdo \t"+
            "Angulo FLEX derecho \t Angulo FLEX izquierdo\n");
    }
    File.AppendAllText(path5, content5);
}
```

Figura 6.9 Funció “excelHABDUP”

Fent referència a les dades que comporta el fitxer: la primera columna amb el temps, tres amb la posició x, y, z de l'espatlla dreta, tres per la posició de l'espatlla esquerra, sis amb la posició dels dos colzes, sis amb les posicions dels dos canells. Llavors s'insereixen les dades dels dos punts de la columna, el coll, el cap i, finalment, els quatre angles, és a dir, tant el dret com l'esquerre d'abducció i els corresponents per a l'angle de flexió.

Però no es pot afegir contingut si no es posseeix la drecera amb el fitxer creat. És per això que s'introdueix una nova condició, que estableix que si la drecera anterior no existeix (per tant el fitxer no existeix) en creï una de nova, el nou fitxer tindrà la data de realització al capdamunt i, una filera més avall, totes les columnes que anteriorment ja s'han anomenat (coordenades dels punts del cos i angles). Aquesta condició és fonamental ja que si estigués al pas anterior per cada instant de valors es crearia un nou fitxer Excel.

Un cop creat el fitxer i determinats tots els valors cal inserir-los de la següent manera: "File.AppendAllText" (nom de la drecera, contingut a inserir).

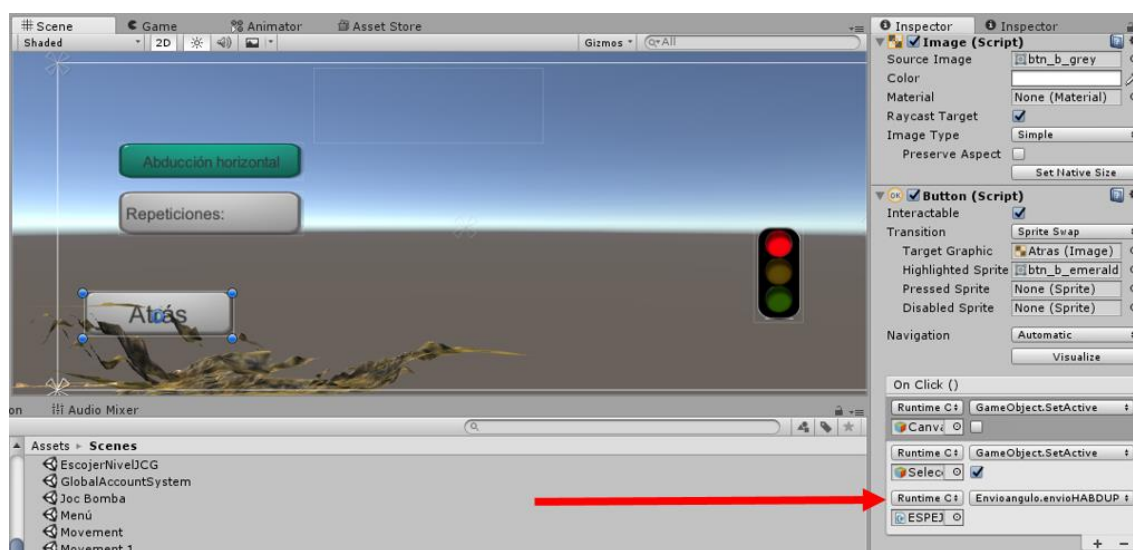


Figura 6.10 Funcionalitat botó "Atrás"

Per altra banda i per finalitzar amb l'exercici d'abducció horitzontal, el botó "Atrás" del canvas de l'exercici d'abducció horitzontal té una altra funcionalitat a part de tornar al menú de selecció de moviments (és a dir, desactivar el canvas actual i activar el canvas del menú de selecció). Aquest activa una funció anomenada "envioHABDUP", que és dins del codi "Envioangulo", que al prémer el botó del canvas s'estaran enviant les dades pertinents a la base de dades de MySQL.

```

public void envioHABDUP()
{
    System.DateTime dateTime = System.DateTime.Now;
    fecha = dateTime.ToString("dd-MM-yyyy");
    //hora
    hora = System.DateTime.Now.Hour;
    minutos = System.DateTime.Now.Minute;
    Hora = hora + ":" + minutos;
    //DNI
    DNI = GameObject.FindGameObjectsWithTag("DNI");
    nombre = DNI[1].name;

    Movimiento = "HABDUP";
    angulo = "-";
    angulo2 = "-";
    Simetrianum = "-";

    Nivell = "-";
    Bombtime = "-";

    repHABDUP = BodySourceUpperBody.repHABDUP;

    HABDUptime = BodySourceUpperBody.HABDUptime.ToString(); //bs.HABDUptime.ToString();
    StartCoroutine(AngleInsert(nombre, angulo, angulo2, fecha, Hora, Movimiento, Simetrianum, repHABDUP, HABDUptime, Nivell, Bombtime));
}

```

Figura 6.11 Funció "envioHABDUP"

Aquesta funció (Figura 6.11) defineix la data, el DNI del pacient i el nom del exercici. Tot seguit es dona un valor nul als valors que corresponen a altres exercicis (amb un guió) i finalment es reben des del codi anterior els valors "repHABDUP" i "HABDUptime", que corresponen a les repeticions totals realitzades i el temps total de l'exercici. A continuació, es duen aquestes dades a una altra corutina anomenada "AngleInsert", que és comuna per a tots els exercicis.

```

IEnumerator AngleInsert(string nombre, string angulo, string
{
    WWWForm form = new WWWForm();

    form.AddField("DNIPost", nombre);
    form.AddField("AnguloMaxPost", angulo);
    form.AddField("AnguloMax2Post", angulo2);
    form.AddField("FechaPost", fecha);
    form.AddField("HoraPost", Hora);
    form.AddField("MovimientoPost", Movimiento);
    form.AddField("SimetriaPost", Simetrianum);
    form.AddField("RepHABDUPPost", RepHABDUP);
    form.AddField("HABDUptimePost", HABDUptime);
    form.AddField("NivellPost", Nivell);
    form.AddField("BombtimePost", Bombtime);

    WWW www = new WWW(AngleInsertURL, form);
    yield return www;

    //Mensaje.text = www.text;
}

```

Figura 6.12 Corutina "AngleInsert"

Aquesta corutina (Figura 6.12) envia les dades a un fitxer PHP anomenat "AngleInsert.php", que és l'encarregat d'establir la connexió directament amb la base de dades.



```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName = "Users";
7
8
9  //////////////////////////////////////
10 //
11 //////////////////////////////////////
12
13 //Variable from the patient
14 $Nombre = $_POST["DNIPost"];
15 $Angulo=$_POST["AnguloMaxPost"];
16 $Angulo2=$_POST["AnguloMax2Post"];
17 $Fecha= $_POST["FechaPost"];
18 $Hora=$_POST["HoraPost"];
19 $Movimiento=$_POST["MovimientoPost"];
20 $Simetria=$_POST["SimetriaPost"];
21 $RepHABDUP=$_POST["RepHABDUPPost"];
22 $HABDUptime=$_POST["HABDUptimePost"];
23 $Nivell=$_POST["NivellPost"];
24 $Bombtime=$_POST["BombtimePost"];
25
26
27
28
29
30 //Make Connection
31 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
32 //Check Connection
33 if(!$conn){
34     die("Connection Failed. ". mysqli_connect_error());
35 }
36 //
37
38 $sql= "SELECT PatientName FROM pacientes WHERE pacientes.DNI='".$_Nombre.'";
39
40
41 $sql = "INSERT INTO angulos (DNIP, AngulomaxDerecha, AngulomaxIzquierda, PatientName, Dia, Hora,Ejercicio, Simetria) SELECT pacientes.DNI,
42 '".$_Angulo.'', '".$_Angulo2.'', pacientes.PatientName, '".$_Fecha.'', '".$_Hora.'', '".$_Movimiento.'', '".$_Simetria.'" FROM pacientes WHERE pacientes.DNI='".$_Nombre.'";
43 $sql = "INSERT INTO angulos (DNIP, AngulomaxDerecha, AngulomaxIzquierda, PatientName, Dia, Hora,Ejercicio, Simetria, RepeticionesHABDUP, TiempoHABDUP, Nivel,
44 TiempoBomba) SELECT pacientes.DNI, '".$_Angulo.'', '".$_Angulo2.'', pacientes.PatientName, '".$_Fecha.'', '".$_Hora.'', '".$_Movimiento.'', '".$_Simetria.'', '".$_RepHABDUP.'',
45 '".$_HABDUptime.'', '".$_Nivell.'', '".$_Bombtime.'" FROM pacientes WHERE pacientes.DNI='".$_Nombre.'";
46
47 $result = mysqli_query($conn, $sql);
48
49 if(!result) echo "there was an error";
50 else echo "Everything ok.";
51
52 ?>

```

Figura 6.13 Script "AngleInsert.php"

La Figura 6.13 mostra el codi PHP. En el cas d'aquest projecte només es van canviar les línies de codi del número 13 al 24, que és on se li aporta la dada, i les línies 43 a 45 (estan separades per a que es vegi a la captura, però realment en són una de sola), que indiquen que les dades que s'han aportat en les línies anteriors corresponen a certa columna de la taula de la base de dades.

## 6.2 Flexió braços (desactivació bomba)

Com en l'exercici d'abducció horitzontal, hi ha un botó en el menú de selecció de moviments on s'activen dues funcions del fitxer "canviaEscena": la funció "canviaBomba" i la funció "playbomb".



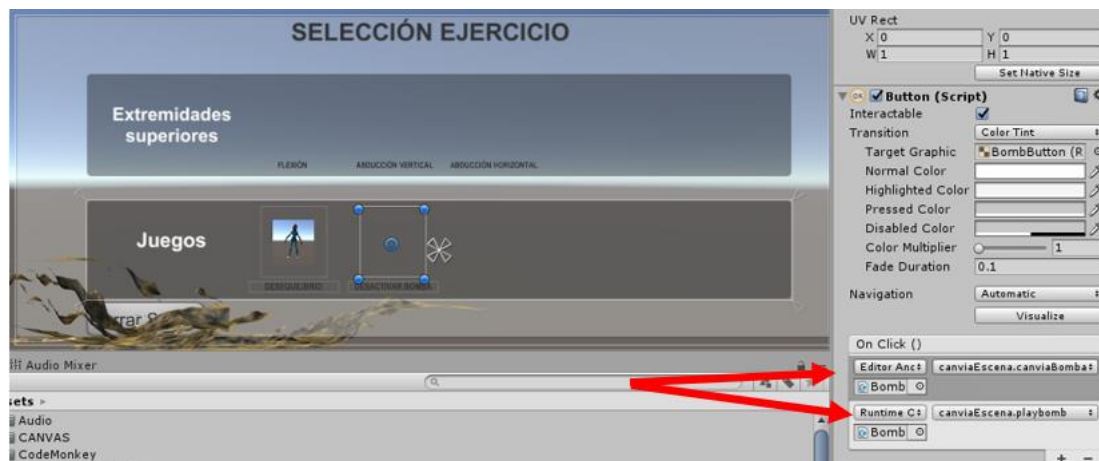


Figura 6.14 Funcionalitats botó joc desactivació bomba

```

public void canviaBomba()
{
    SceneManager.LoadScene("Joc Bomba");
}

public void playbomb()
{
    bomba.PlayBomb = 1;
    bomba.Bombtime = 0;
}

```

Figura 6.15 Funció "canviaBomba" i "playBomb"

La funció "canviaBomba" només té la finalitat de carregar l'escena que tingui el nom "Joc Bomba", que és on realment es troba el joc. La funció "playbomb" iguala la variable "PlayBomb" a 1 (quan la bomba explota s'iguala a 0) i iguala la variable "Bombtime" a 0 (aquesta fa la mateixa funció que "HABDUptime", que aporta el temps durant el transcurs del joc).

```

x = Head.transform.position.x;
y = Head.transform.position.y;
z = Head.transform.position.z;
xbuttonR = x + 0.4f;
xbuttonL = x - 0.4f;
zbuttons = z + 0.2f ;
yCam = y + 1;
zCam = z - 4f;

pushbuttonL.transform.position = new Vector3(xbuttonL, ybuttons, zbuttons);
pushbuttonR.transform.position = new Vector3(xbuttonR, ybuttons, zbuttons);
Cam.transform.position = new Vector3(x, yCam, zCam);

```

Figura 6.16 Variables a "Update"

Dins el codi "bomba" (que s'activa al entrar en l'escena del joc, ja que aquest codi està inserit en un objecte de l'escena), que és el principal del joc, primer s'inicialitza la variable "Nivell" amb el valor de 0. Dins de la funció "Update", s'iguala la variable "y" a la component y de la posició del cap, i s'igualen "xbuttonR", "xbuttonL" i "zbuttons" als seus valors adients, aquests valors són la posició en l'eix X i Z dels pulsadors. Per altra banda, també es modifica els valors "yCam" i "zCam", que corresponen a les

components y i z de la càmera. A continuació, es modifica la posició dels *Game Objects* corresponents, tant del botó dret, com de l'esquerre així com el de la càmera.

Dins de la funció *"Update"*, situada dins el codi *"bomba"* existeix un seguit de variables i condicions interessants d'analitzar. En el primer dels casos hi ha la variable *"Bombtime"*, que ja anteriorment ha estat inicialitzada en 0. Aquesta variable porta el compte del temps total de realització de l'exercici. Tot seguit, es troba una condició que estableix que si els dos botons no són actius, el compte enrere prendrà un valor igual a 99 (aquest fet es produeix en les transicions entre nivells). Tanmateix, s'aprecien dues condicions que fan referència a si el compte enrere és major o menor a 9.5 segons. Aquestes dues condicions donen el nombre adequat de números en el requadre del temps (ja que els valors superiors a 10 tenen 2 xifres i hi hauria una incoherència). Finalment, cal remarcar una última condició, la qual si s'indica que el compte enrere és inferior a 3 segons i la bomba encara no ha explotat, s'activi un efecte de fum. Tot i que està representat a la Figura 6.17, aquesta funció no fa res, ja que la intenció era introduir aquest efecte en l'entorn del joc, però a causa de la manca de temps no es va arribar a finalitzar.

```
void Update () {
    Bombtime += Time.deltaTime;

    if (!pushbuttonR.gameObject.active && !pushbuttonL.gameObject.active)
    {
        countdown = 99;
    }

    if (countdown >= 9.5f)
    {
        countdowntext.text = ("00 : " + countdown.ToString("f0"));
    }

    if (countdown < 9.5f)
    {
        countdowntext.text = ("00 : 0" + countdown.ToString("f0"));
    }
    if (countdown < 3f && explotat==false)
    {
        //BombSmoke();
    }
}
```

Figura 6.17 Variables i condicions dins d'"Update"

Novament dins d'"Update", es té una nova condició que si *"Nivell"* té un valor igual a 0 i la component de text del *Game Object* *"level"* és igual a 1, la seva funció serà mostrar al *canvas* del joc aquesta informació. Més endavant es defineix *"ybuttons"*, que és l'alçada a la que es troben els botons (a la

mateixa del cap), que es defineix amb la variable “y”. Tot seguit s’activa el *canvas* adient (que no té cronòmetre) i es desactiva el del cronòmetre (Figura 6.18).

```
if (Nivell == 0)
{
    level.text = "1";
    ybuttons = y*1.0f;
    CanvasChrono.SetActive(false);
    Canvas.SetActive(true);
}
```

Figura 6.18 Condició “Nivell == 0”

Com ja se sap, a l’iniciar el joc l’usuari es troba en el primer nivell, i per avançar en els següents cal contactar amb els 2 botons. Això no seria possible sense l’ajuda dels *colliders*, que són geometries que fan certa acció al contactar entre ells, a la figura Figura 6.19 es poden observar els *colliders* de les dues mans en qüestió i un dels polsadors, aquests *colliders* son definits per els prismes transparents amb arestes de color verd.

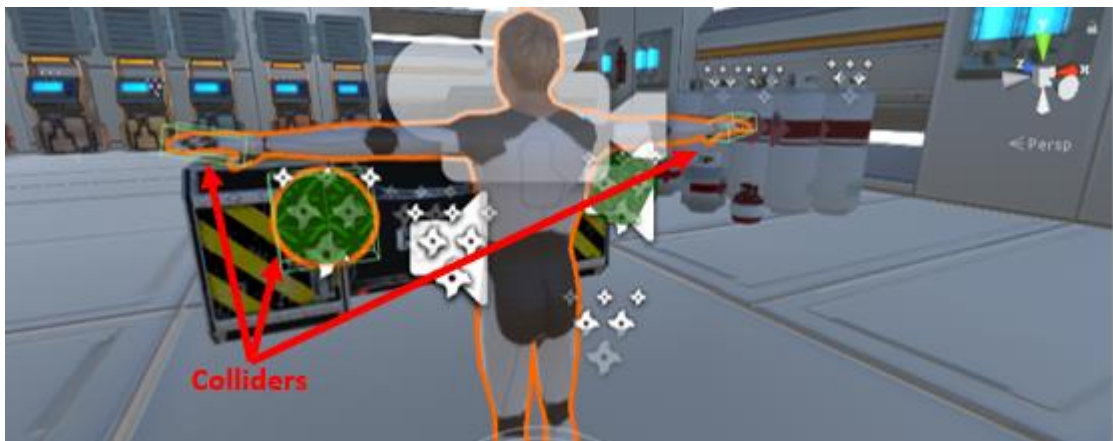


Figura 6.19 Colliders mans i polsadors

Dins dels 2 polsadors hi ha inserit un codi (el mateix per als 2) anomenat “pushbutton” (Figura 6.20), el qual té dues funcions principals. Si es comença pel final, la funció “OnCollisionEnter”, ja predefinida per Unity3D, s’activarà a l’inici d’una col·lisió, és a dir, quan un objecte topi contra un altre. Aleshores, dins d’aquesta funció primerament s’augmenta el valor de la variable “numero” en 1 (aquesta variable és la que fa desactivar la bomba). En segon lloc, es duu a terme la funció “Desactivate” i, per finalitzar,

desactiva el polsador per a que desaparegui. Per altra banda, la funció “Desactivate” que està just a sobre, activa un petit sistema de partícules.

```

}

void Desactivate()
{
    var exp = desactivate.GetComponent<ParticleSystem>();
    exp.Play();
}

void OnCollisionEnter(Collision other)
{
    //Destroy(this.gameObject);
    pushbutton_R.SendMessage("Button");
    numero = numero + 1;
    desactivate.transform.position = this.gameObject.transform.position;
    Desactivate();
    this.gameObject.active = false;
}

```

Figura 6.20 Script “pushbutton”

Un cop aclarida la temàtica dels botons, cal tornar al codi anterior (“bomba”). A la Figura 6.21, s’observa una condició on si el valor de la variable “numero” del codi “pushbutton” és igual a 2, s’activa una corutina anomenada “Temporizador”. Una vegada aquesta finalitza, es duu a terme la funció “BombDesactivation” i es torna a donar a la variable “numero” un valor igual a 0 per al següent nivell.

```

if (pushbutton.numero == 2)
{
    StartCoroutine(Temporizador());

    BombDesactivation();
    pushbutton.numero = 0;
}

```

Figura 6.21 Condició “pushbutton.numero == 2”

La corutina “Temporizador” espera 3 segons, aleshores s’augmenta el valor de “Nivell” en 1, torna a activar els 2 polsadors per a que siguin visibles i dona un valor de 15 a la variable “countdown”, que s’explicarà més endavant.

Per altra banda, la funció “BombDesactivation” busca, primerament, el *Game Object* amb el nom “Desactivation” (el qual és un sistema de partícules per a la desactivació de la bomba) i, tot seguit, l’executa. Finalment es reproduïx un so confirmant que la bomba ha estat neutralitzada.

```

IEnumerator Temporizador()
{
    yield return new WaitForSeconds(3);
    Nivell = Nivell + 1;
    pushbuttonR.gameObject.active = true;
    pushbuttonL.gameObject.active = true;
    Bombaa.gameObject.active = true;
    countdown = 15;
}

void BombDesactivation()
{
    Desactivation = GameObject.Find("Desactivation");
    var expl = Desactivation.GetComponent<ParticleSystem>();
    expl.Play();
    neutralizada_sound.SendMessage("BombOut");
}

```

Figura 6.22 Corutina "Temporizador" i Figura XX Funcio "BombDesactivation"

Un cop s'avança de nivell s'entra en una altra condició, que activa el *canvas* que s'havia desactivat en el nivell 0 i desactiva el que havia estat desactivat. D'aquesta manera, es mostra un *canvas* amb cronòmetre que permet al pacient veure el compte enrere per a l'explosió. En aquest cas s'utilitza la variable "countdown", que té un compte enrere de 15 segons, ja que s'havia definit previamente. Pel que fa a l'alçada, aquesta segueix sent la mateixa, ja que es va trobar convenient fer un nivell per a que el pacient es pogués fer a la idea de l'alçada en que es trobaria els polsadors i, llavors, fer-ne un segon amb la mateixa alçada però amb compte enrere, i així fins un transcurs de 8 nivells.

```

if (Nivell == 1)
{
    CanvasChrono.SetActive(true);
    Canvas.SetActive(false);
    levelCountdown.text = ("2");

    countdown -= Time.deltaTime;
    ybuttons = y * 1.0f;
}

```

Figura 6.23 Condició "Nivell == 1"

Novament dins d'"Update" del codi "bomba", es troba una altra condició: en el cas que el valor de "countdown" sigui igual o inferior a 0, el valor de "countdown" s'igualarà a 0, i si la variable booleana "explotat" és falsa (i, per tant, encara no ha explotat la bomba), s'executarà la funció "Explode" i, d'aquesta manera, "explotat" serà vertadera (per indicar que la bomba ja ha explotat i que no torni a explotar encara que el compte enrere sigui igual a 0).

```

if (countdown <= 0)
{
    countdown = 0;
    if (explotat == false)
    { Explode();
      explotat = true;
    }
}

```

Figura 6.24 Condició "countdown &lt;= 0"

Referent a la funció “Explode”, en primer lloc es desactiva el *Game Object* “Bombaa”, el qual és la representació de la bomba que es pot visualitzar en el joc. Tanmateix, es busca el *Game Object* amb nom “BombExplosion” (*Game Object* amb sistema de partícules en forma d’explosió) i s’activa la component del sistema de partícules d’aquell mateix *Game Object*. Seguidament, la variable “explotat” es torna certa i es reproduïx el so relacionat amb el sistema de partícules anteriorment ja activat; d’aquesta manera tots els efectes ja estaran activats. A continuació, s’activa la corutina “TempGameOver” i s’igual a 0 la variable “PlayBomb”.

```
void Explode()
{
    Bombaa.gameObject.active = false;
    BombExplosion = GameObject.Find("BombExplosion");
    var exp = BombExplosion.GetComponent<ParticleSystem>();
    exp.Play();
    explotat = true;
    explosion_sound.SendMessage("Boom");
    StartCoroutine(TempGameOver());
    PlayBomb = 0;
}
```

Figura 6.25 Funció “Explode”

Per a entendre correctament la funció anterior “Explode”, s’ha de veure la corutina cridada “TempGameOver”. Aquesta corutina espera 2 segons (per a que es vegi correctament l’efecte d’explosió) i activa un *canvas*, anomenat “GameOver”, destinat únicament a la finalització del joc en forma de derrota, és a dir, la bomba ha explotat i, per tant, no s’ha aconseguit superar o assolir fins l’últim nivell.

```
IEnumerator TempGameOver()
{
    yield return new WaitForSeconds(2);
    GameOver.SetActive(true);
}
```

Figura 6.26 Corutina “TempGameOver”

Com es pot veure a la Figura 6.27, el *canvas* “GameOver” només té un botó, el qual activa dues funcions del codi “canviaEscena” i un altre a “Envioangulo”.



Figura 6.27 Canvas "GameOver"

Dins del codi "canviaEscena" s'han creat dos funcions a l'hora de clicar sobre el botó per a retrocedir: "backMenu" i "quitbomb". La primera d'elles carrega l'escena 3, que és la de selecció d'exercici (també es podria haver fet introduint el nom de l'escena, com s'ha realitzat en altres ocasions). Per altra banda, "quitbomb" iguala novament la variable "PlayBomb" a 0.

```
public void backMenu()
{
    SceneManager.LoadScene(3);
}

public void quitbomb()
{
    bomba.PlayBomb = 0;
}
```

Figura 6.28 Funció "backMenu" i funció "quitbomb"

L'altra funció que activa el canvas "GameOver" és "envioBOMBA" dins del codi "Envioangulo" (Figura 6.29). Aquesta és molt semblant a la funció anteriorment vista "envioHABDUP", que obté la data i el DNI del pacient corresponent, assigna valors nuls a les dades d'altres exercicis per a evitar confusions i adjudica al joc la variable "Movimiento", per definir de quin exercici es tracta, que s'assigna com a "BOMBA", i després se li assignen els valors tant del nivell al que ha arribat el pacient com el temps total de realització

```

public void envioBOMBA()
{
    System.DateTime dateTime = System.DateTime.Now;
    fecha = dateTime.ToString("dd-MM-yyyy");
    //hora
    hora = System.DateTime.Now.Hour;
    minutos = System.DateTime.Now.Minute;
    Hora = hora + ":" + minutos;
    //DNI
    DNI = GameObject.FindGameObjectsWithTag("DNI");
    nombre = DNI[1].name;

    Movimiento = "BOMBA";
    angulo = "-";
    angulo2 = "-";
    Simetrianum = "-";

    Nivell = bomba.Nivell.ToString();//(bomba.Nivell.ToString());
    Bombtime = bomba.Bombtime.ToString();//bomba.Bombtime.ToString();

    repHABDUP = "-";

    HABDUptime = "-"; ;//bs.HABDUptime.ToString();
    StartCoroutine(AngleInsert(nombre, angulo, angulo2, fecha, Hora, Movimiento, Simetrianum, repHABDUP, HABDUptime, Nivell, Bombtime));
}

```

Figura 6.29 Funció “envioBomba”

Fins ara s’ha explicat què fa el programa si es perd. A continuació s’explica el procés quan es superen els 8 nivells del joc.

Per començar, es té una condició dins la funció “Update” (novament dins el codi “bomba), el qual si la variable “Nivell” és igual a 8 (cal recordar que es començava des d’un nivell 0; per tant, aquest seria un hipotètic nivell 9) s’activa el *canvas* anomenat “Win”(Figura 6.31). És un *canvas* especialment dissenyat per a la victòria de l’usuari, que com succeeix en el cas de derrota, només té un botó i serveix per a tornar al menú de selecció de moviment.

```

if (Nivell == 8)
{
    Win.SetActive(true);
}

```

Figura 6.30 Condició “Nivell == 8”



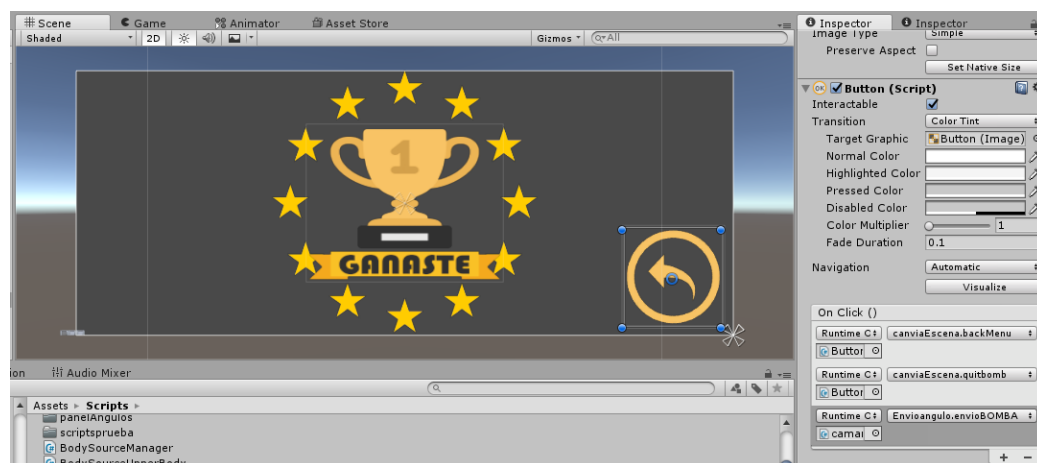


Figura 6.31 Canvas "Win"

## 7. Planificació del projecte

### 7.1 Pla de treball

Aquest treball, està relacionat amb un altre realitzat en el mateix període per dos autors diferents (Aplicación de la realidad virtual a la rehabilitación), que han desenvolupat uns exercicis diferents que van conjuntament a la mateixa aplicació. Cal destacar que al principi del projecte els dos grups van treballar conjuntament, degut a que les tasques encomanades repercutien a ambdós, però després cada un es va separar per dissenyar els nous exercicis de rehabilitació.

En primer lloc, es va partir d'una sèrie de vídeos introductoris ("Máster en Programación de Videojuegos con Unity 2018 y C#" de la plataforma d'aprenentatge Udemy i "Unity VR Fundamentals" de la dissenyadora de videojocs MJ Johns) que van servir per a la **familiarització** de les funcions que es tenia en l'**entorn de Unity3D** i la **programació amb C#** d'una forma bàsica.

Seguidament, es va fer una **reunió amb els quatre alumnes del projecte i els tres docents** per tal de donar les primeres pautes, per marcar uns objectius inicials i començar a dividir el treball. Aquests objectius eren **exportar les dades ja fos a cada frame** o després d'un cert temps, portar tot el projecte anterior al software de control de versions Git i unir els dos projectes anteriors, és a dir, unir els exercicis de les extremitats inferiors a l'entorn del de les superiors. Finalment, es van exportar certes dades (posicions de les articulacions de les extremitats superiors del cos i els angles d'abducció i flexió) en un fitxer Excel cada 0.5 segons.

Un cop finalitzada la primera fase es va realitzar una nova **reunió** amb les mateixes persones de l'anterior junt **amb la fisioterapeuta** i voluntària de l'associació ADFO Cristina Molas. En aquesta reunió es van definir els propers passos per al projecte seguint les seves necessitats. Es va decidir crear un nou exercici d'abducció horitzontal i, a més a més, afegir 2 jocs: un per tal que els pacients aixequin els braços (flexió-extensió i abducció-adducció) i un altre per avaluar el desequilibri del cos. És aquí on els camins dels 2 grups es van separar.

Es va treballar en l'**exercici d'abducció horitzontal**, on primerament es va **crear un entorn virtual** semblant al dels altres exercicis. A continuació, se li va **crear a la monitora virtual** (seria l'avatar de la fisioterapeuta) l'**animació** (moviments que s'atorguen a un avatar; en aquest cas serien d'abducció horitzontal de braços) i, tot seguit, es van implementar mètodes per a l'**obtenció dels punts del cos** necessaris i **angles** per a poder exportar-los posteriorment. Més tard, es va **exporta aquestes dades al fitxer Excel** amb les dades del pacient (DNI, exercici, data de realització i les dades de l'exercici corresponent) i, a continuació, es va **implementar un comptador de repeticions**. Finalment, es va **exportar el número de repeticions i temps total a la base de dades**.

Una vegada es va acabar amb la creació de l'exercici d'abducció horitzontal, es va decidir que el joc es desenvolupés en un ambient futurista o espacial i que el seu principal objectiu fos desactivar una bomba abans de cert temps; si no fos així, la bomba explotaria. Per a que el pacient pogués desactivar aquesta bomba virtual, es posarien 2 polsadors a una certa alçada del cap i així es realitzaria l'exercici d'aixecar els braços.

Ja sabent la temàtica del futur joc, es va començar **ideant les mecàniques del joc** amb més profunditat per a que la programació fos més senzilla. Seguidament, es va procedir, igual que a l'exercici, a **definir angles i punts del cos per a exportar-los al fitxer extern** i, altre cop igual que en l'exercici, es va **exportar el temps i el nivell** al que arriba el pacient **a la base de dades principal**. Un cop tot això estava acabat i, per tant, el joc era operatiu, es va començar a buscar objectes per a la **decoració de l'ambient**, animacions per a la bomba i l'ambient i la millora i/o creació d'elements informatius o bé funcionals del que vindrien a ser menús del joc.

Un cop finalitzada l'aplicació, es van **millorar certs errors**, com per exemple, canviar la metodologia de càlcul dels angles a un entorn 3D.

Just en el moment en que es va modificar aquesta funció, es van **agafar mostres de certes persones** realitzant tant l'exercici com el joc i així **poder extreure dades reals** i poder-les interpretar.

En la següent figura (Figura 7.1Figura 7.1) es pot veure en forma d'esquema el que seria el resum de la metodologia emprada per a la planificació d'aquest projecte.

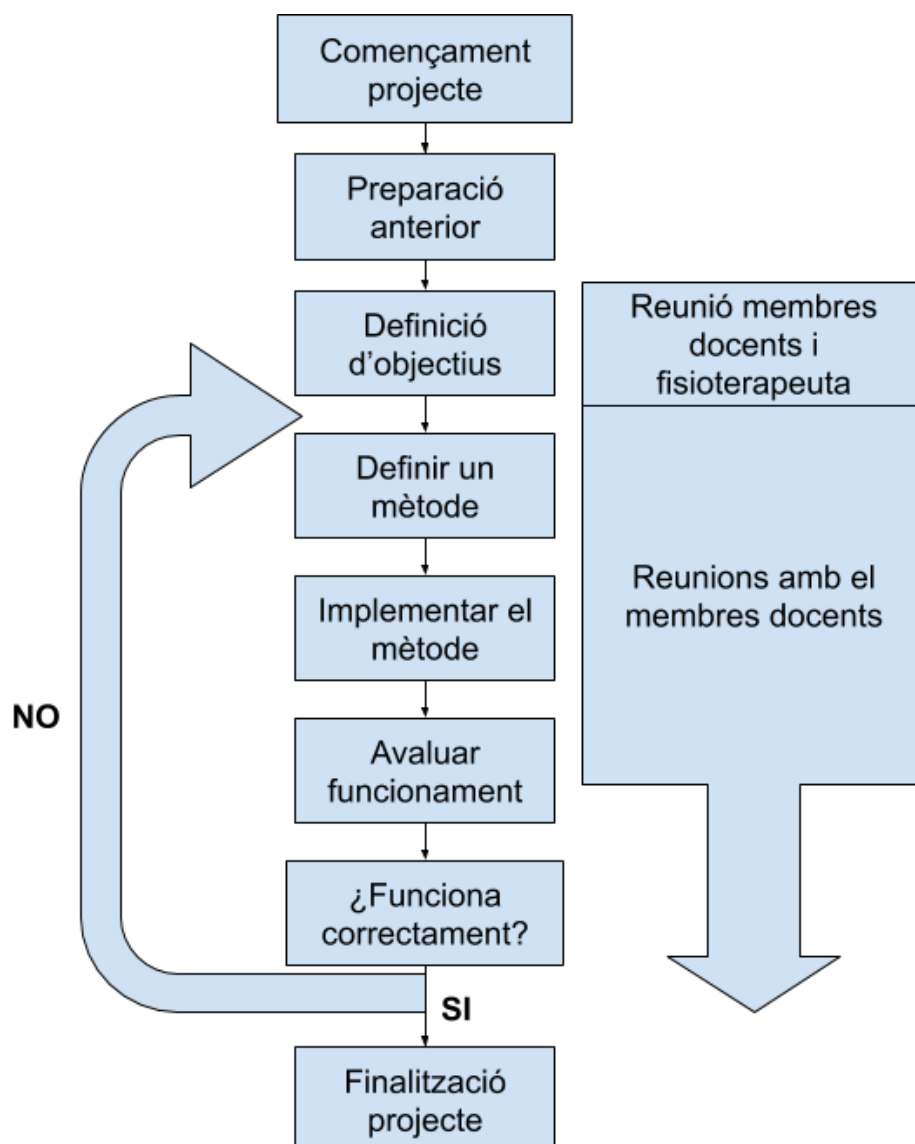
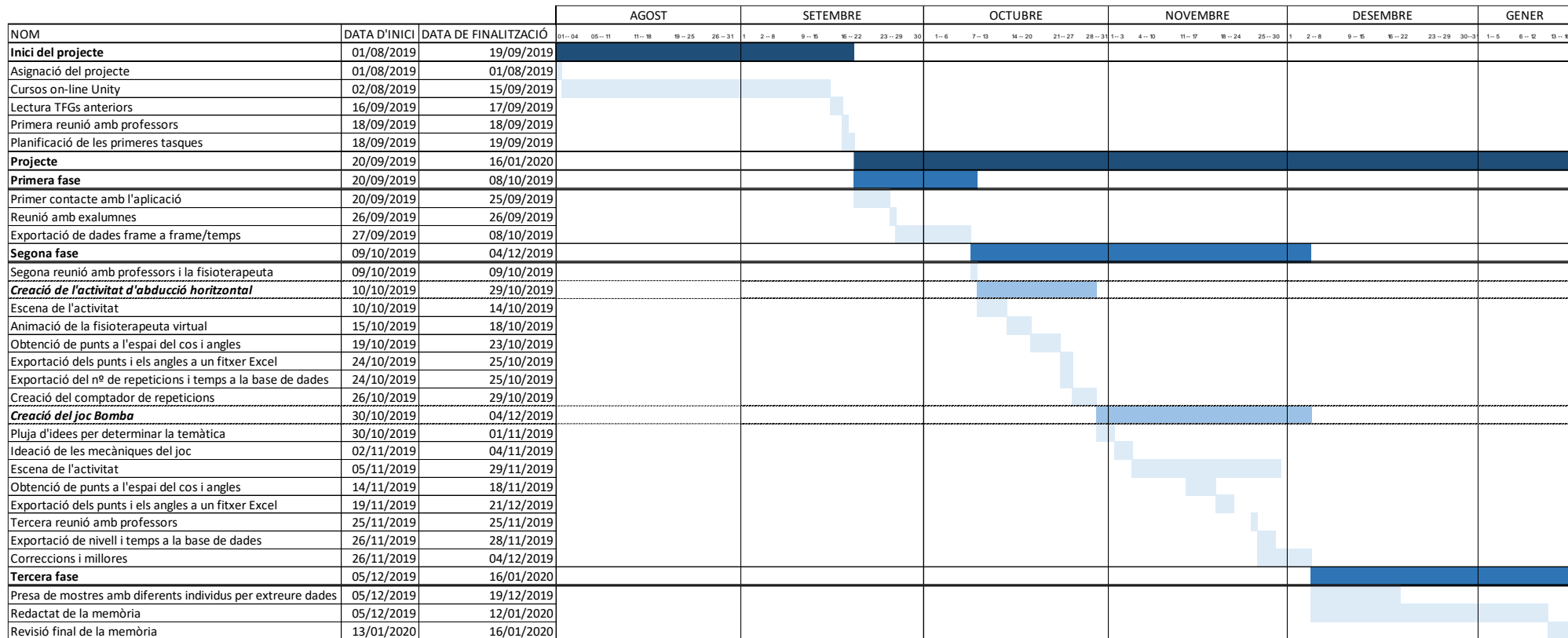


Figura 7.1 Metodologia per a la realització del treball

## 7.2 Diagrama de Gantt

A continuació es mostra el diagrama de Gantt corresponent al projecte, on es mostra tot el procés de creació, així com les tasques que es van realitzar com les reunions que es van tenir amb els professors.



## 7.3 Requeriments funcionals i no funcionals

### Requeriments funcionals

Els requeriments funcionals són declaracions dels serveis que prestarà el sistema en la forma en què reaccionarà a determinats inputs [38]. Quan es parla de les entrades, no necessàriament es parla només de les entrades dels usuaris. Poden ser interaccions amb altres sistemes, respostes automàtiques, processos pre-definits. En alguns casos, els requeriments funcionals dels sistemes també estableixen explícitament el que el sistema no ha de fer. És important recordar que un requeriment funcional pot ser també una declaració negativa, sempre i quan el resultat del seu comportament sigui una resposta funcional a l'usuari a un altre sistema.

### Requeriments no funcionals

Es tracta de requeriments que no es refereixen directament a les funcions específiques subministrades pel sistema (característiques d'usuari), sinó a les propietats de sistema: rendiment, seguretat, disponibilitat. En paraules més senzilles, no parlen de "el que" fa el sistema, sinó de "com" ho fa. Alternativament, defineixen restriccions de sistema com ara la capacitat dels dispositius d'entrada / sortida i la representació de les dades utilitzades en la interfície de sistema.

A continuació es mostra la Taula 7.1 amb els principals requeriments d'aquest projecte.

Requeriments funcionals	Requeriments no funcionals
Exportar de les extremitats superiors que ens dona Kinect	Rendiment bo de l'aplicació, amb un temps de resposta acceptable
Exportar 2 angles, el d'abducció i el de flexió en un àmbit 3D	Utilitzant Kinect i compatible amb HTC Vive
Pujar certs valors a la base de dades	Que no tingui errors que puguin afectar a l'experiència del pacient
Nou exercici (abducció horitzontal)	Entorn de l'exercici igual que els altres exercicis
Nou joc (per a aixecar els braços)	Entorn del joc futurista
Accés a les dades exportades	El joc tractarà de desactivar una bomba
	Estímuls auditius al joc per a fer més realista l'escena

Que els 2 menús estiguin connectats i, per tant, es pugui anar del principal al del joc i viceversa i el mateix amb l'exercici
Instruccions per a la correcta realització de l'exercici
Instruccions per a poder jugar correctament al joc
Dades classificades corresponentment (DNI, data i exercici)

*Taula 7.1 Requeriments funcionals i no funcionals*

## 8. Assajos amb voluntaris

Per a extreure resultats i conclusions dels mateixos s'ha fet un seguit d'assajos amb voluntaris. Aquests voluntaris són persones no lesionades, és a dir, no s'ha fet assajos amb pacients reals amb problemes físics i/o cognitius.

Cal destacar que els participants han realitzat aquests assajos sense les ulleres HTC Vive, ja que es va donar l'opció als voluntaris i van decidir dur a terme els exercicis sense les ulleres.

Remarcar també que els voluntaris van realitzar els exercicis sense cap tipus de dificultat.

A continuació, es mostra una taula informativa on s'indica l'edat, el pes, el sexe i l'alçada dels voluntaris que han realitzat, amb molta amabilitat, els exercicis que es presenten en aquest projecte:

	Sexe	Edat	Pes (kg)	Alçada (cm)
<b>Voluntari 1</b>	Masculí	22	84.8	180
<b>Voluntari 2</b>	Masculí	22	75.3	170
<b>Voluntari 3</b>	Masculí	24	70.4	177.5
<b>Voluntari 4</b>	Masculí	29	87.2	183

*Taula 8.1. Informació general dels voluntaris*

Per a poder analitzar les dades més rellevants s'ha decidit crear dos gràfics per a cada voluntari i una taula amb certs valors estadístics. Per una banda, el primer gràfic representa l'evolució de l'angle d'abducció respecte al temps en tots dos braços alhora i el segon gràfic es representa el mateix la mateixa informació però amb l'angle de flexió. Per altra banda, a la taula s'ha calculat el número total de mostres, la mitjana, la desviació estàndard (representada amb  $\sigma$ ), la mediana, el valor mínim, el valor màxim, el número de màxims relatius (anomenats en el document com a pics, per a fer una simplificació), el número total de mínims relatius (mencionats en el document com a valls, per la mateixa raó que els pics) i el coeficient de correlació entre les dues corbes (en els casos testejats, no és un valor de molta utilitat, però en pacients reals pot arribar a ser determinant). Per a aquests mínims i màxims relatius s'ha calculat també la mitjana, desviació estàndard, mínims i màxims corresponents.

En aquesta secció s'analitzen els resultats obtinguts en els exercicis d'abducció horitzontal:



## 8.1 Abducció Horitzontal

### Voluntari 1

El primer fet important a destacar, que es pot observar visualment sense les dades estadístiques, és que el subjecte en el moment de màxima obertura dels braços té tendència a obrir més el braç esquerre (línia taronja a les gràfiques) i, al contrari, en el moment de tancar els braços, que correspon a les valls de les gràfiques, hi ha tendència amb aquest mateix braç esquerre a no finalitzar el moviment.

Una altra dada visual és que els pics en la gràfica de flexió (Taula 8.2) tenen unes valls, en alguns casos, bastant pronunciades (s'atribueix aquest fet al que s'ha anomenat com "efecte rebot"). Però és del cert que si es realitza l'exercici a l'obrir els braços, s'executa un petit retrocés en el moment final.

A les valls, és a dir, el moment del tancament, també es té una irregularitat en els dos braços, encara que no és tan pronunciada com la dels pics de l'angle de flexió.

Un cop es tenen els càlculs estadístics (

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	184.00	184.00	184.00	184.00
<b>Mitjana</b>	59.44	63.37	132.25	134.22
<b>Coefficient de correlació</b>	0.9814		0.9620	
<b>Desviació estàndard (<math>\sigma</math>)</b>	33.57	32.62	27.77	27.27
<b>Mínim</b>	2.74	11.32	88.34	85.91
<b>Mediana</b>	66.51	67.50	126.43	126.14
<b>Màxim</b>	120.27	125.39	174.12	178.49
<b>Número de pics</b>	13.00	17.00	23.00	21.00
<b>Mitjana pics</b>	81.09	76.53	137.21	142.59
<b><math>\sigma</math> pics</b>	36.12	34.95	31.87	30.52

<b>Mínim pics</b>	2.74	19.02	88.34	101.28
<b>Màxim pics</b>	113.68	119.92	170.69	173.42
<b>Mediana pics</b>	94.16	76.58	160.88	164.31
<b>Número de valls</b>	12.00	16.00	22.00	21.00
<b>Mitjana valls</b>	32.47	42.85	119.97	123.38
<b><math>\sigma</math> valls</b>	30.98	25.55	28.07	25.57
<b>Mínim valls</b>	4.08	16.65	89.82	101.21
<b>Màxim valls</b>	80.49	79.65	169.21	174.40
<b>Mediana valls</b>	14.77	27.90	106.13	113.05

**¡Error! No se encuentra el origen de la referencia.)** , es confirma el que s'ha vist anteriorment: la mitjana de les valls és bastant superior al braç esquerra comparat amb el dret. Per altra banda, en els pics sí que es confirma el que s'ha dit anteriorment sobre l'alçada d'aquests, però només en el cas de la flexió. Per a l'abducció aquesta dada no és tan indicativa, ja que si s'observa a la part inferior hi ha varies pujades, és per això que es tenen pics en el moment del tancament dels braços, i comporta que la mitjana no aporti tanta informació com en els altres casos.

Es pot veure que els valors de la desviació estàndard són bastant similars. Per tant, el rang de moviment és semblant en ambdós casos, ja que el que un braç s'obri més que l'altre després es compensa en el moment del tancament.

### Evolució de l'angle d'abducció respecte al temps

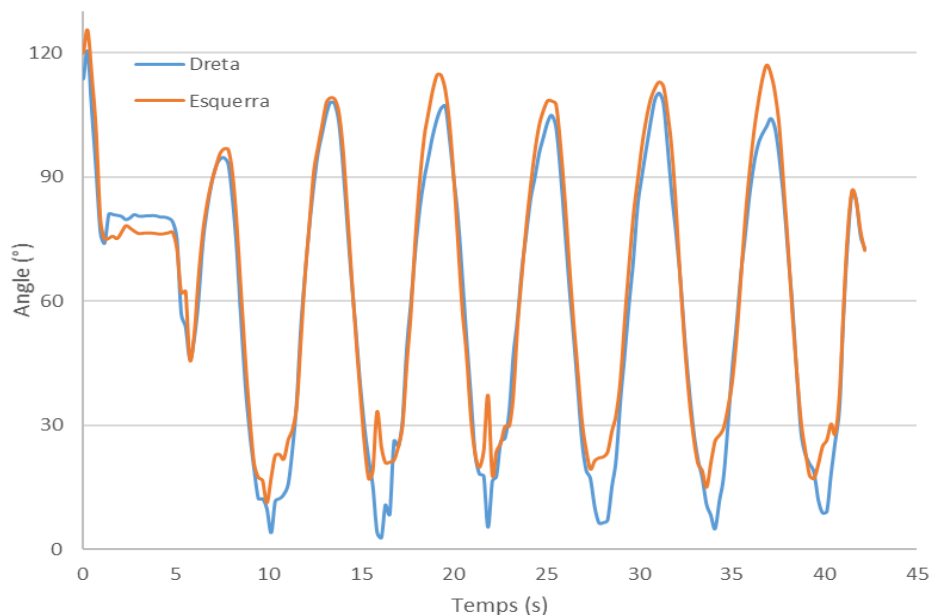


Figura 8.1 Abducció voluntari 1 (abducció horitzontal)

### Evolució de l'angle de flexió respecte al temps

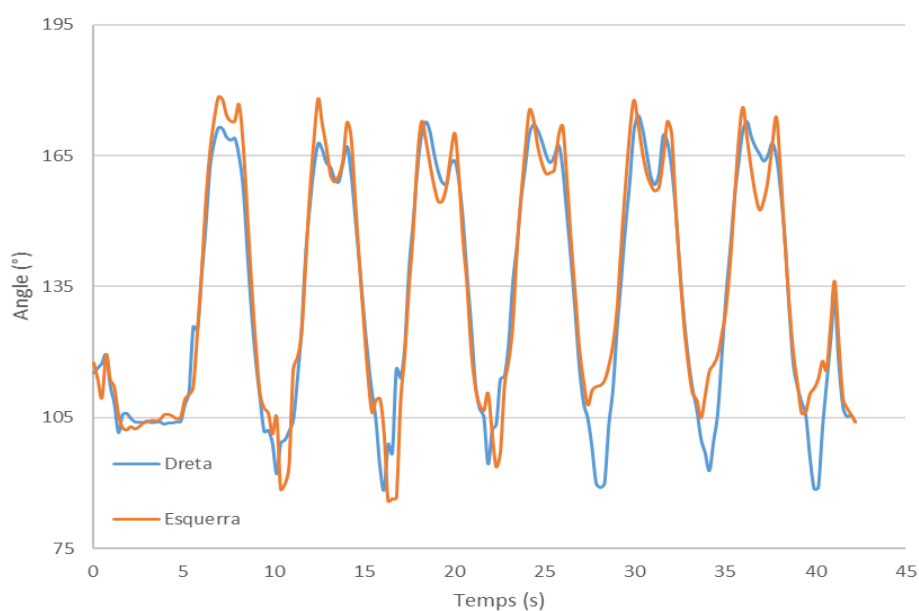


Figura 8.2 Flexió voluntari 1 (abducció horitzontal)

Abducció Horitzontal			Flexió	
	Dreta	Esquerra	Dreta	Esquerra
n	184.00	184.00	184.00	184.00
Mitjana	59.44	63.37	132.25	134.22
Coefficient de correlació	0.9814		0.9620	
Desviació estàndard ( $\sigma$ )	33.57	32.62	27.77	27.27
Mínim	2.74	11.32	88.34	85.91
Mediana	66.51	67.50	126.43	126.14
Màxim	120.27	125.39	174.12	178.49
Número de pics	13.00	17.00	23.00	21.00
Mitjana pics	81.09	76.53	137.21	142.59
$\sigma$ pics	36.12	34.95	31.87	30.52
Mínim pics	2.74	19.02	88.34	101.28
Màxim pics	113.68	119.92	170.69	173.42
Mediana pics	94.16	76.58	160.88	164.31
Número de valls	12.00	16.00	22.00	21.00
Mitjana valls	32.47	42.85	119.97	123.38
$\sigma$ valls	30.98	25.55	28.07	25.57
Mínim valls	4.08	16.65	89.82	101.21
Màxim valls	80.49	79.65	169.21	174.40
Mediana valls	14.77	27.90	106.13	113.05

Taula 8.2 Dades estadístiques voluntari 1 (abducció horitzontal)

## Voluntari 2

En aquest cas s'observa visualment que els moviments en els 2 braços són més semblants. També es pot apreciar la davallada a la part superior de la gràfica de flexió (com succeïa amb el voluntari 1). Un altre fet interessant és l'apreciació del suposat cansament del voluntari 2; això es pot veure a la gràfica de flexió, que cada cop van disminuint els màxims. Per altra banda, es pot observar unes alteracions en la senyal al final de l'exercici; amb això s'interpreta que el voluntari volia parar al arribar a un cert nombre de repeticions o que es va tancar l'objectiu del sensor Kinect en el moment de finalitzar la prova.

Evolució de l'angle d'abducció respecte

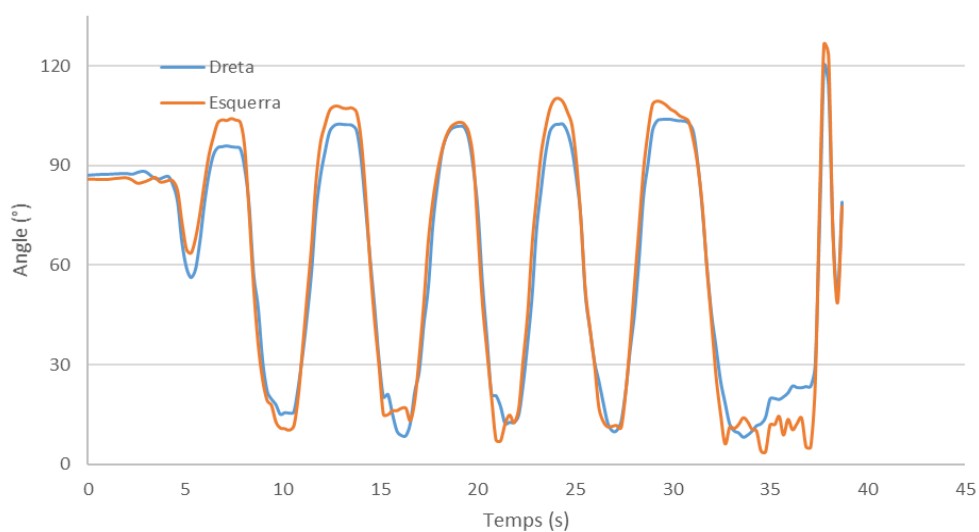


Figura 8.3 Abducció voluntari 2 (abducció horitzontal)

Evolució de l'angle de flexió respecte al temps

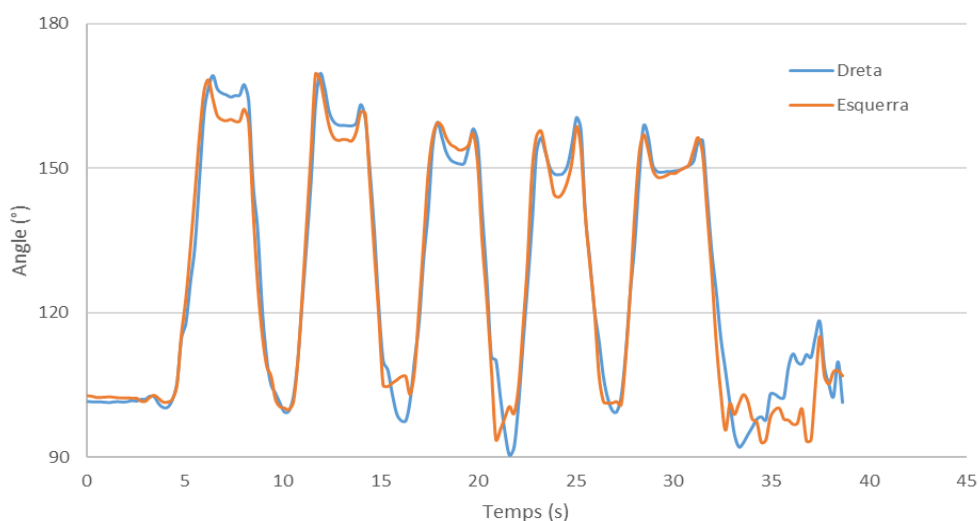


Figura 8.4 Flexió voluntari 2 (abducció horitzontal)

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	170.00	170.00	170.00	170.00
<b>Mitjana</b>	80.11	78.28	111.01	110.93
<b>Coefficient de correlació</b>	0.9692		0.8997	
<b>Desviació estàndard (<math>\sigma</math>)</b>	16.34	14.20	11.74	12.69
<b>Mínim</b>	41.86	45.09	100.22	98.75
<b>Mediana</b>	83.86	81.38	104.33	104.01
<b>Màxim</b>	124.45	124.13	144.67	142.84
<b>Número de pics</b>	34.00	34.00	34.00	32.00
<b>Mitjana pics</b>	83.22	79.49	112.16	111.28
<b><math>\sigma</math> pics</b>	12.19	10.32	11.20	12.50
<b>Mínim pics</b>	43.49	48.08	100.30	99.65
<b>Màxim pics</b>	96.94	90.01	133.75	137.57
<b>Mediana pics</b>	84.22	81.35	105.80	103.89
<b>Número de valls</b>	34.00	34.00	34.00	32.00
<b>Mitjana valls</b>	76.54	75.10	109.26	108.02
<b><math>\sigma</math> valls</b>	18.29	13.47	9.83	10.17
<b>Mínim valls</b>	45.06	49.19	100.88	100.12
<b>Màxim valls</b>	111.93	91.28	133.66	136.45
<b>Mediana valls</b>	84.06	81.38	103.47	103.65

Taula 8.3 Dades estadístiques voluntari 2 (abducció horitzontal)

### Voluntari 3

En primer lloc, es pot percebre que aquest voluntari va fer més repeticions que els anteriors. També mostra una regularitat que els altres no havien mostrat (principalment en la gràfica de flexió. Tot i que també es pot veure que la desviació estàndard és inferior a la dels altres voluntaris; aquest fet indica que el rang de moviment no és tan ampli com en els altres casos (si es miren els màxims i mínims es reafirma aquesta hipòtesi).

Tanmateix, és important destacar que aquest subjecte no presenta unes baixades tan pronunciades en l'angle de flexió al arribar a la màxima obertura.

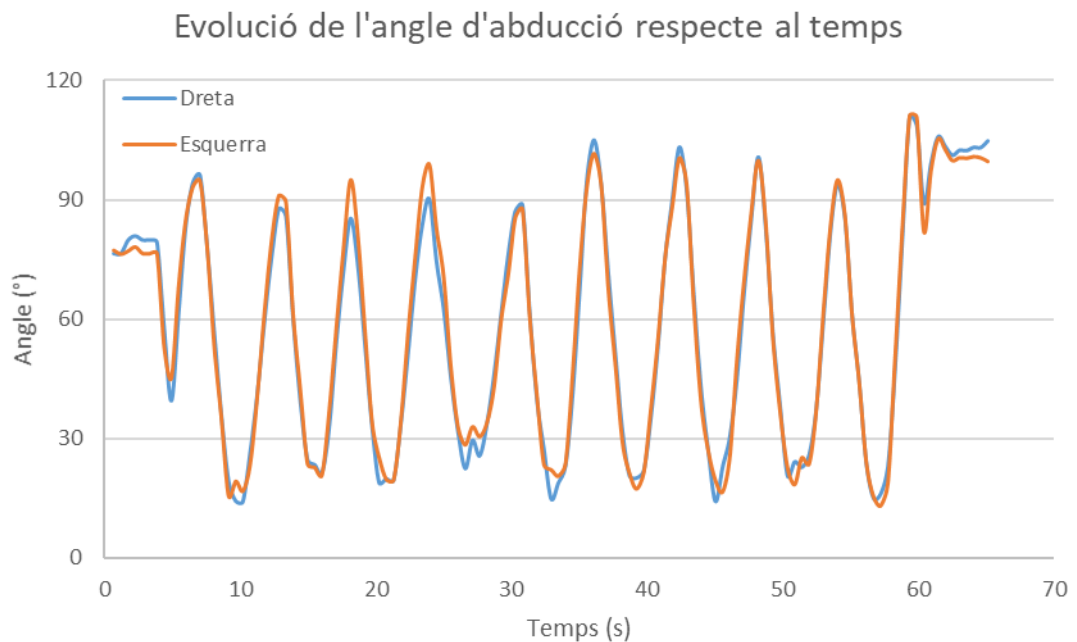


Figura 8.5 Abducció voluntari 3 (abducció horitzontal)

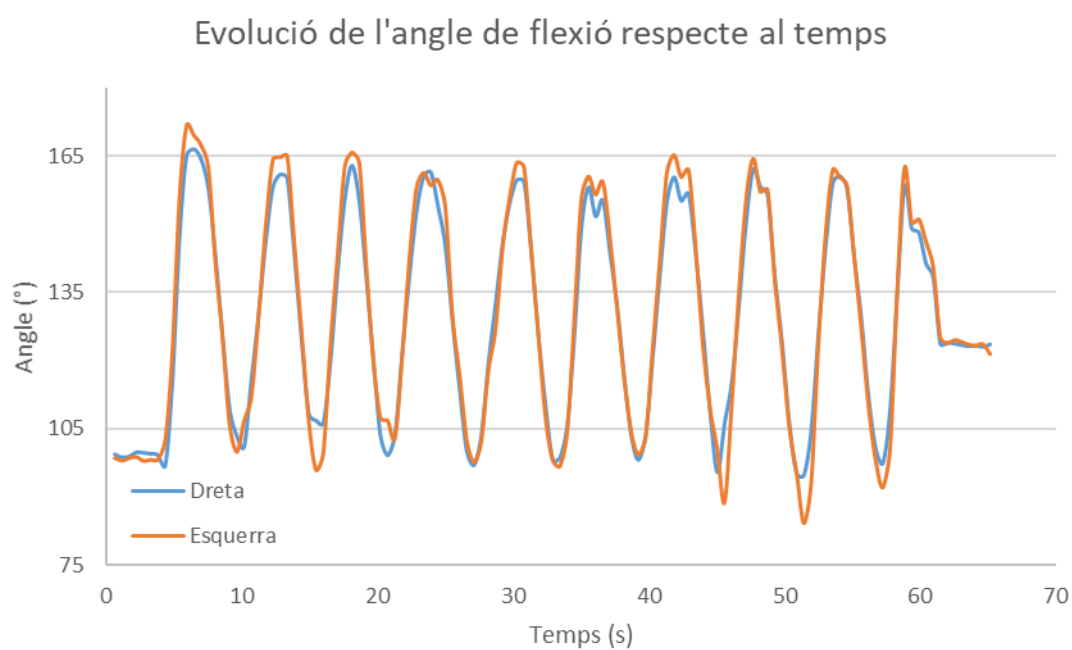


Figura 8.6 Flexió voluntari 3 (abducció horitzontal)



	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	123.00	123.00	123.00	123.00
<b>Mitjana</b>	58.17	58.75	129.42	130.24
<b>Coefficient de correlació</b>	0.9927		0.9894	
<b>Desviació estàndard (<math>\sigma</math>)</b>	29.56	29.46	22.60	24.93
<b>Mínim</b>	14.10	13.09	94.75	84.10
<b>Mediana</b>	59.64	59.62	125.68	126.37
<b>Màxim</b>	110.93	111.08	166.58	171.78
<b>Número de pics</b>	18.00	18.00	16.00	18.00
<b>Mitjana pics</b>	77.39	76.53	144.87	148.99
<b><math>\sigma</math> pics</b>	27.66	26.78	18.54	17.58
<b>Mínim pics</b>	18.87	15.61	98.93	98.47
<b>Màxim pics</b>	103.19	100.48	165.22	164.70
<b>Mediana pics</b>	84.26	84.73	153.29	156.29
<b>Número de valls</b>	17.00	18.00	16.00	18.00
<b>Mitjana valls</b>	46.36	44.99	114.65	120.90
<b><math>\sigma</math> valls</b>	34.55	33.01	20.74	27.88
<b>Mínim valls</b>	14.20	15.41	95.19	95.07
<b>Màxim valls</b>	108.78	110.78	160.37	165.25
<b>Mediana valls</b>	29.54	29.01	105.14	106.04

Taula 8.4 Dades estadístiques voluntari 3 (abducció horitzontal)

## Voluntari 4

Es pot detectar visualment que hi ha uns canvis de direcció molt bruscos i els pendents són més pronunciats que en els anteriors. Això es deu a la realització de la simulació a una velocitat superior a la que indica la fisioterapeuta virtual; tot i anar massa ràpid, es veu una similitud entre els 2 braços.

Per altra banda, cal destacar que la taula estadística no és gaire indicativa, ja que tan a l'inici com al final de les gràfiques es tenen pertorbacions bastant voluminoses que fan que les dades no siguin de molta utilitat.

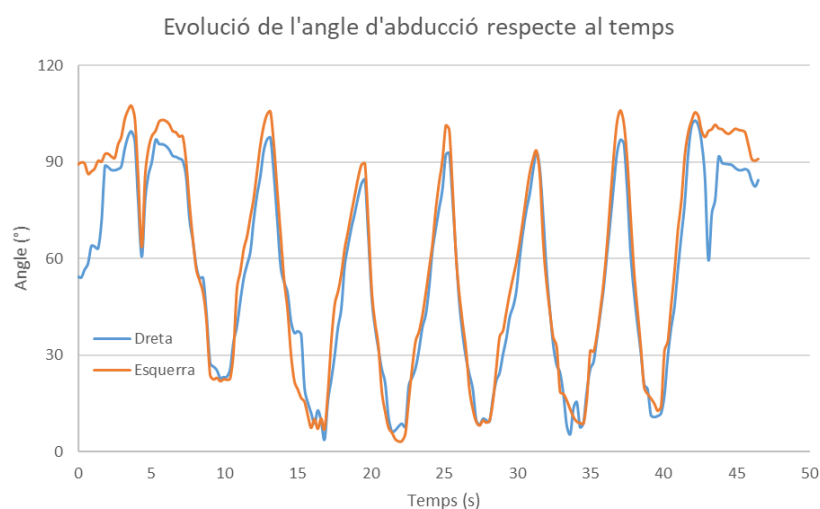


Figura 8.7 Abducció voluntari 4 (abducció horitzontal)

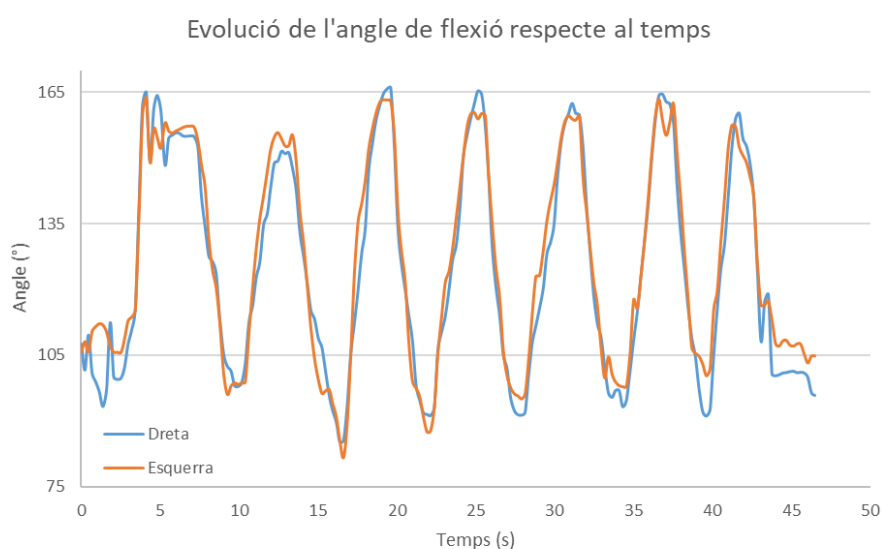


Figura 8.8 Flexió voluntari 4 (abducció horitzontal)

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	203.00	203.00	203.00	203.00
<b>Mitjana</b>	54.48	59.65	124.22	126.77
<b>Coefficient de correlació</b>	0.9753		0.9765	
<b>Desviació estàndard (<math>\sigma</math>)</b>	30.06	34.02	24.53	23.72
<b>Mínim</b>	3.73	3.25	85.11	81.72
<b>Mediana</b>	56.53	62.81	119.95	123.22
<b>Màxim</b>	102.81	107.42	166.09	163.80
<b>Número de pics</b>	19.00	21.00	17.00	24.00
<b>Mitjana pics</b>	65.72	72.87	138.83	135.41
<b><math>\sigma</math> pics</b>	33.88	38.74	28.30	26.31
<b>Mínim pics</b>	7.90	7.16	96.88	96.94
<b>Màxim pics</b>	101.17	106.03	165.80	162.79
<b>Mediana pics</b>	82.34	90.55	154.99	152.29
<b>Número de valls</b>	19.00	20.00	17.00	23.00
<b>Mitjana valls</b>	39.84	46.23	112.85	120.21
<b><math>\sigma</math> valls</b>	32.49	39.95	27.19	26.96
<b>Mínim valls</b>	7.83	3.95	89.59	85.61
<b>Màxim valls</b>	87.44	99.67	164.97	163.80
<b>Mediana valls</b>	25.24	23.40	99.96	107.40

Taula 8.5 Dades estadístiques voluntari 4 (abducció horitzontal)

## 8.2 Flexió braços (desactivació bomba)

Abans d'analitzar els resultats d'aquest joc és important dir que els valors d'aquesta prova no són tan regulars, ja que aquí el temps no el marca la fisioterapeuta virtual, sinó el propi usuari.

### Voluntari 1

Primerament, destacar que no s'aprecia una progressió durant el pas dels nivells, ja que els pics i les valls no tenen diferències d'alçada molt destacables entre un nivell i el següent. Per tant, la distància del polsador amb el cap pot ser que sigui massa prudent, ja que el voluntari fa un moviment semblant i avança nivells sense problemes. També és important manifestar que tant l'aplicació com el joc estan dissenyades per a persones que han patit ictus i potser sí que és adequat per a elles. Per aquesta raó seria beneficiós fer proves amb pacients reals per així poder graduar aquests valors.

En segon lloc destacar que el braç dret, principalment en l'angle de flexió, té una tendència més exagerada i, per tant, obté uns valors més alts.

Per altra banda és interessant destacar que el braç dret té un rang de valors més estret que el dret referent a l'abducció, però referent a la flexió és totalment el contrari, que es pot confirmar gràcies a la desviació estàndard.

Evolució de l'angle d'abducció en funció al temps

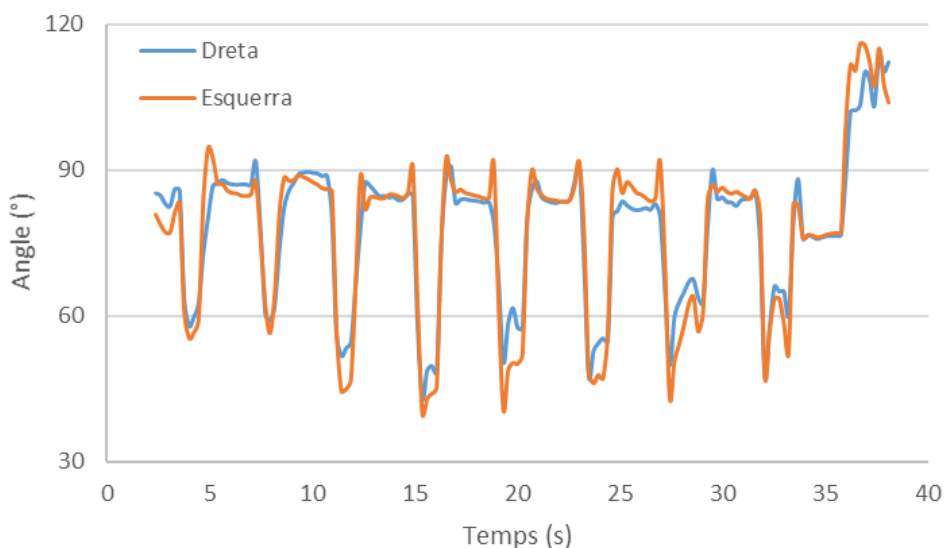


Figura 8.9 Abducció voluntari 1 (bomba)

## Evolució de l'angle de flexió en funció al temps

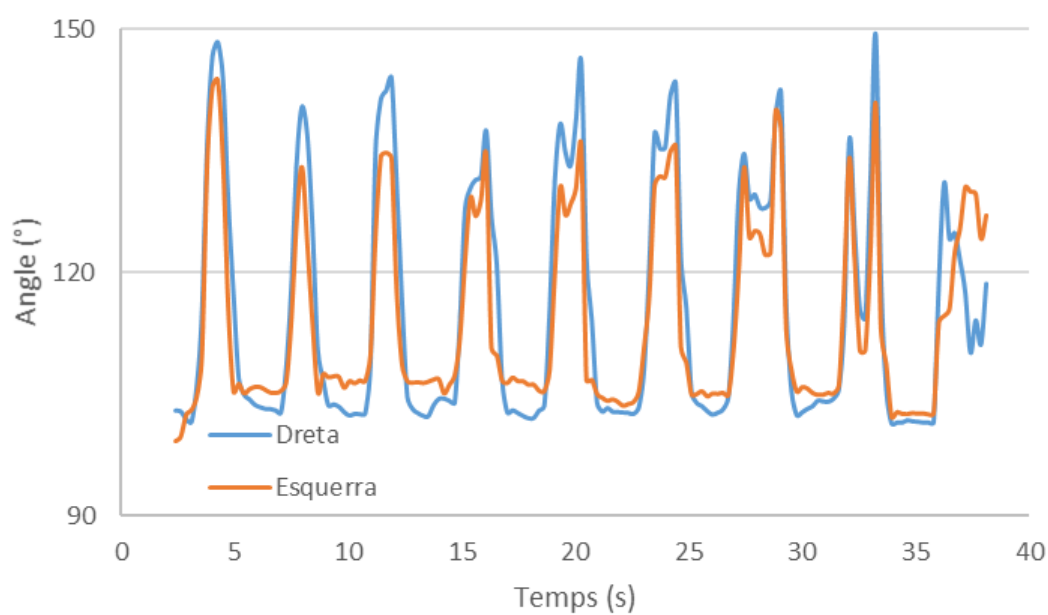


Figura 8.10 Flexió voluntari 1 (bomba)

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	155.00	155.00	155.00	155.00
<b>Mitjana</b>	77.74	77.65	115.18	113.73
<b>Coefficient de correlació</b>	0.9650		0.9245	
<b>Desviació estàndard (<math>\sigma</math>)</b>	14.34	16.86	14.49	11.54
<b>Mínim</b>	43.02	39.75	101.34	99.11
<b>Mediana</b>	82.97	84.16	106.78	107.01
<b>Màxim</b>	113.78	115.95	149.22	143.79
<b>Número de pics</b>	32.00	30.00	26.00	36.00
<b>Mitjana pics</b>	81.43	80.40	119.90	114.73
<b><math>\sigma</math> pics</b>	13.12	14.58	15.65	11.75
<b>Mínim pics</b>	48.66	46.14	101.34	102.11
<b>Màxim pics</b>	110.13	110.38	146.54	142.91
<b>Mediana pics</b>	83.89	84.27	119.71	106.67
<b>Número de valls</b>	32.00	30.00	26.00	35.00
<b>Mitjana valls</b>	77.05	76.81	110.52	111.71
<b><math>\sigma</math> valls</b>	15.06	16.47	12.07	10.37
<b>Mínim valls</b>	49.65	47.83	101.46	102.49
<b>Màxim valls</b>	113.78	112.31	137.08	135.64
<b>Mediana valls</b>	82.05	83.15	104.11	106.60

Taula 8.6 Dades estadístiques voluntari 1 (bomba)

## Voluntari 2

En primer lloc ressaltar les grans pertorbacions que es tenen tan en el final com en l'inici del joc. Com en l'anterior voluntari, aquest ha superat tots els nivells sense dificultat, tot i que en l'últim nivell es veu una lleugera diferència amb les altres valls si es parla de la flexió (només en el braç dret), així com també existeix una exageració dels moviments en el braç dret, tant en flexió com abducció, encara que la desviació estàndard no ho demostra a causa d'aquestes grans interferències.

Evolució de l'angle d'abducció en funció al temps

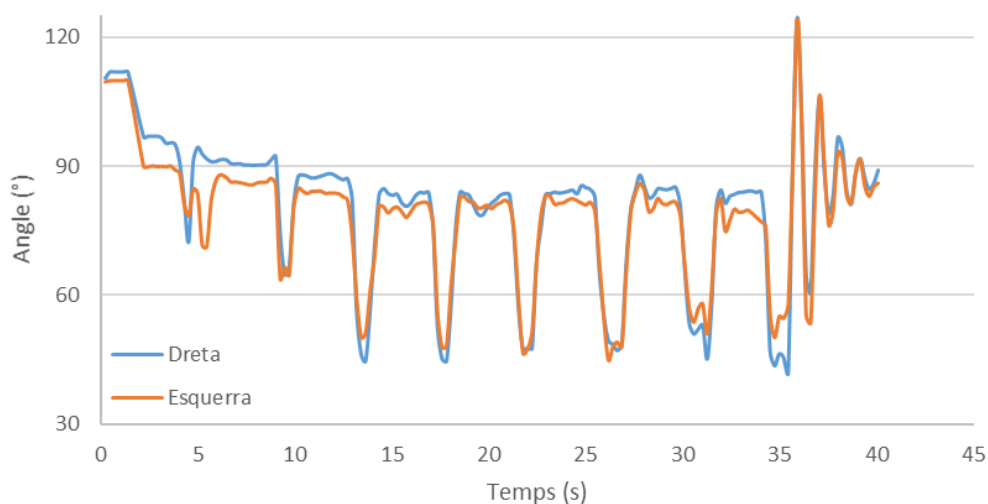


Figura 8.11 Abducció voluntari 2 (bomba)

Evolució de l'angle de flexió en funció al temps

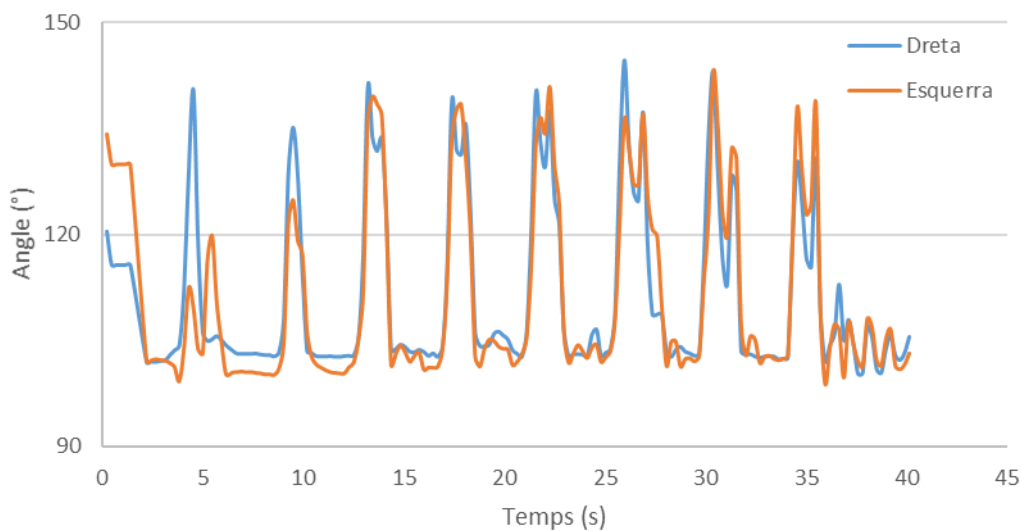


Figura 8.12 Flexió voluntari 2 (bomba)

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	170.00	170.00	170.00	170.00
<b>Mitjana</b>	80.11	78.28	111.01	110.93
<b>Coefficient de correlació</b>	0.9692		0.8997	
<b>Desviació estàndard (<math>\sigma</math>)</b>	16.34	14.20	11.74	12.69
<b>Mínim</b>	41.86	45.09	100.22	98.75
<b>Mediana</b>	83.86	81.38	104.33	104.01
<b>Màxim</b>	124.45	124.13	144.67	142.84
<b>Número de pics</b>	34.00	34.00	34.00	32.00
<b>Mitjana pics</b>	83.22	79.49	112.16	111.28
<b><math>\sigma</math> pics</b>	12.19	10.32	11.20	12.50
<b>Mínim pics</b>	43.49	48.08	100.30	99.65
<b>Màxim pics</b>	96.94	90.01	133.75	137.57
<b>Mediana pics</b>	84.22	81.35	105.80	103.89
<b>Número de valls</b>	34.00	34.00	34.00	32.00
<b>Mitjana valls</b>	76.54	75.10	109.26	108.02
<b><math>\sigma</math> valls</b>	18.29	13.47	9.83	10.17
<b>Mínim valls</b>	45.06	49.19	100.88	100.12
<b>Màxim valls</b>	111.93	91.28	133.66	136.45
<b>Mediana valls</b>	84.06	81.38	103.47	103.65

Taula 8.7 Dades estadístiques voluntari 2 (bomba)



### Voluntari 3

Aquest voluntari va realitzar la prova, però per un error de programació les seves dades no es van desar.

### Voluntari 4

Com ocorre en l'exercici anterior, l'individu ha realitzat la prova amb suma brusquedat, és per aquesta raó que es veuen unes pendents molt abruptes i unes diferències bastant destacables entre ambdós braços en tots dos angles; es pot confirmar altra vegada amb la desviació estàndard. A l'abducció, l'esquerra té un marc més gran de moviment i, en canvi, a la flexió és al revés

És important dir que en aquest cas sí que es veu una necessitat d'aixecar més el braç a mesura que avancen els nivells, això es pot veure a la gràfica d'abducció.

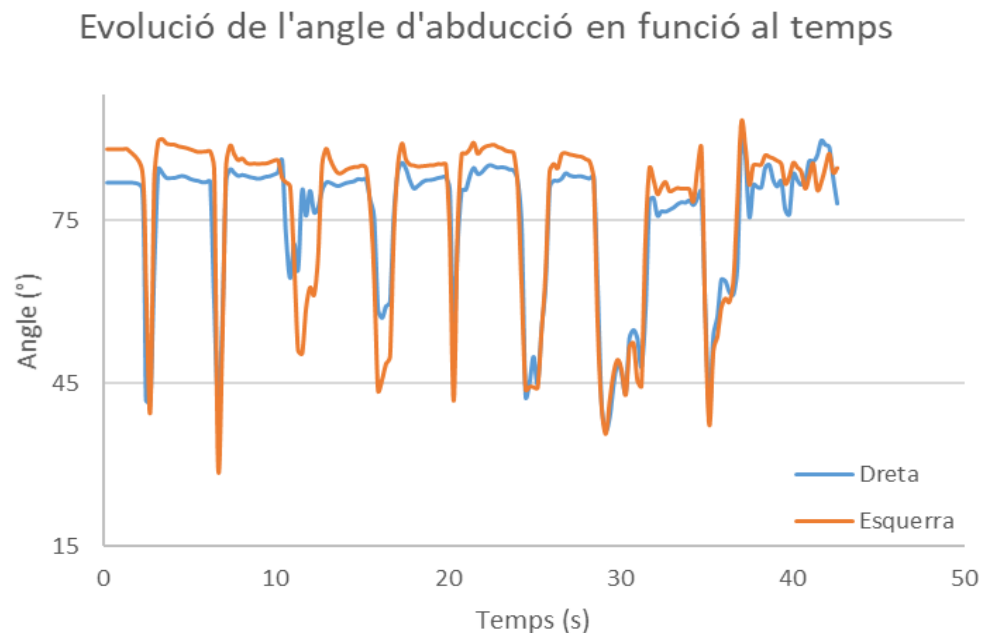


Figura 8.13 Abducció voluntari 4 (bomba)

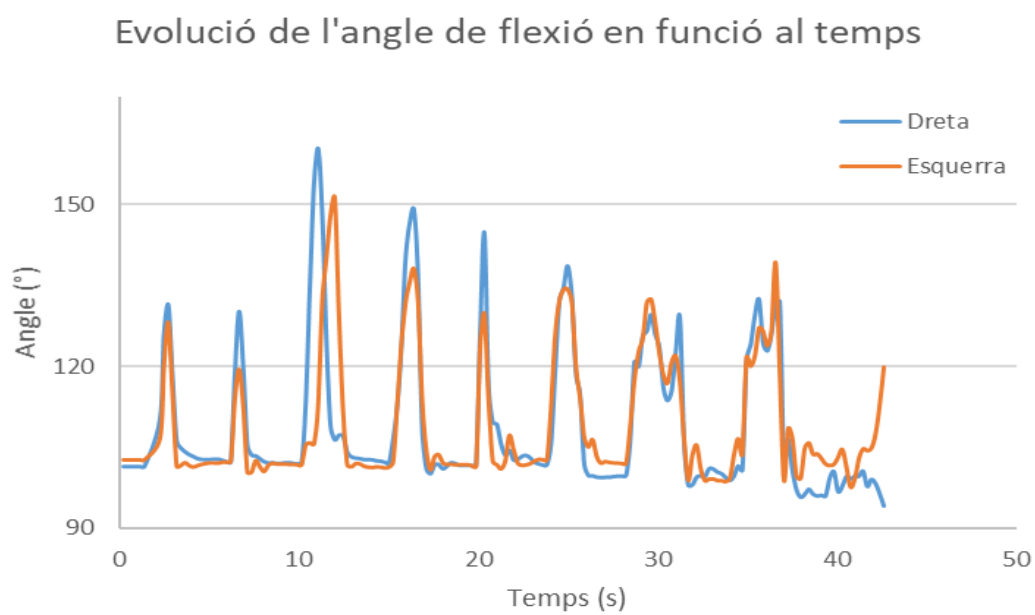


Figura 8.14 Flexió voluntari 4 (bomba)

	Abducció Horitzontal		Flexió	
	Dreta	Esquerra	Dreta	Esquerra
<b>n</b>	181.00	181.00	181.00	181.00
<b>Mitjana</b>	75.26	76.94	108.93	108.99
<b>Coefficient de correlació</b>	0.9172		0.7591	
<b>Desviació estàndard (<math>\sigma</math>)</b>	12.97	15.57	13.30	11.51
<b>Mínim</b>	36.36	29.56	94.20	97.52
<b>Mediana</b>	81.82	84.81	102.78	102.60
<b>Màxim</b>	89.54	93.09	160.47	151.28
<b>Número de pics</b>	37.00	30.00	31.00	34.00
<b>Mitjana pics</b>	76.29	78.90	110.87	110.12
<b><math>\sigma</math> pics</b>	10.93	12.91	15.94	12.83
<b>Mínim pics</b>	44.94	43.85	96.06	98.83
<b>Màxim pics</b>	86.85	89.61	153.82	148.19
<b>Mediana pics</b>	81.83	84.55	102.32	103.51
<b>Número de valls</b>	37.00	30.00	30.00	33.00
<b>Mitjana valls</b>	73.81	73.53	104.43	105.53
<b><math>\sigma</math> valls</b>	13.34	15.90	8.28	6.90
<b>Mínim valls</b>	40.67	40.26	96.21	98.73
<b>Màxim valls</b>	86.95	89.36	132.10	126.49
<b>Mediana valls</b>	80.34	82.38	102.04	102.03

Taula 8.8 Dades estadístiques voluntari 4 (bomba)

## 9. Anàlisi de l'impacte ambiental

Es pot declarar que l'impacte ambiental d'aquest projecte és pràcticament nul. Principalment, es pot dividir aquest impacte en 2 parts: la primera fa referència al consum elèctric de tots els dispositius utilitzats (ordinador, monitor, sensor de moviment Kinect, l'adaptador del Kinect i les ulleres HTC Vive), que és un consum moderat ja que es pot portar a un àmbit domèstic.

Per altra banda, part d'aquest impacte es deu tant a la creació, com al transport, així com als residus pertinents de tots els dispositius que s'utilitzen.

Tots els aparells electrònics, en general, són de difícil reciclatge, ja que solen contenir metalls pesats i altres elements químics de gran toxicitat, com el liti, el mercuri, l'arsènic... Per aquesta raó, els fabricants prefereixen utilitzar nous materials en comptes d'optar per la via del reciclatge, encara que davant l'escassetat d'aquests s'està investigant cada cop més per poder-los reutilitzar després de la vida útil de l'aparell.

Encara que els fabricants optessin per la via del reciclatge, el volum de gent que porta els seus aparells electrònics a punts verds o deixalleries és molt reduït i, per tant, seria encara més difícil arribar a un model d'economia totalment circular.

A causa d'aquests elements difícils de reciclar, la mala gestió per part dels propietaris, amb addició dels plàstics que porten tots aquests dispositius, pot portar a l'augment de la contaminació d'aigües (tant subterrànies, rius, mars i llacs) i terres [39].

Per poder-ho evitar, es va crear la Directiva 2012/19/UE sobre residus d'aparells elèctrics i electrònics (RAEE) [40].

Aquesta directiva té com a principals objectius:

- Prevenir la creació de residus d'aparells elèctrics i electrònics (RAEE)
- Promoure la reutilització, el reciclatge i altres formes de valoritzar residus d'aparells elèctrics i electrònics (AEE)
- Recolzar l'ús eficaç dels recursos i la recuperació de matèries primes secundàries valuoses.

A causa d'aquesta nova Directiva Europea, es va aplicar a Espanya el Reial Decret 110/2015, el qual permet als consumidors retornar petits aparells a les botigues d'electrònica. Exigeix també a tot aparell elèctric i electrònic de ser marcat amb el símbol d'un contenidor marcat amb una creu (com el de la Figura 9.1), a més de ser recollit i processat, o reciclat, un cop es converteixin en residu.



*Figura 9.1. Símbol per marcar els aparells elèctrics o electrònics [41]*

S'ha comprovat que tots els aparells electrònics utilitzats per a la realització del projecte estiguin etiquetats de la forma correcta i, per tant, tinguin el símbol de la Figura 9.1.

Tot i això, l'impacte ambiental de la tele-rehabilitació posseeix també una part positiva, i és que si s'utilitza de forma habitual, tant pacients com fisioterapeutes evitaran desplaçaments fins a la clínica i/o casa del pacient en qüestió i, per conseqüent, s'evitarà un consum de combustible per part d'aquests.

## 10. Conclusions

En aquest projecte s'ha dissenyat una aplicació de realitat virtual, focalitzada en la millora dels tractaments de rehabilitació dels afectats d'ictus. L'objectiu principal fou, doncs, la creació d'un exercici d'abducció horitzontal i el desenvolupament d'un joc que permet la flexió dels braços; tanmateix, es va testejar l'aplicació amb quatre voluntaris sans (no s'ha pogut fer assajos amb pacients reals, ja que l'ADFO no va poder quadrar durant el quadrimestre una sessió per a tals efectes) amb la intenció d'exportar les dades d'ambdós casos a un fitxer Excel i a la base de dades de phpMyAdmin.

Per altra banda, s'ha topat amb una sèrie de limitacions tècniques que van alentir, en alguns moments, avançar en el treball. Les més destacables són les relacionades amb el sensor de moviment Kinect, que no és capaç de detectar correctament els moviments de la mà, o que el seu funcionament es veu alterat quan durant l'activitat del pacient es troba algun objecte o persona enmig. Per altra banda, cal remarcar que la base de dades no permet la importació de dades en forma matricial i, per tant, s'haurien d'haver passat a un àmbit vectorial, cosa que dificultaria la importació i la interpretació d'aquestes.

Malgrat els inconvenients, l'entusiasme de treballar en un projecte amb una clara finalitat solidària, un dels objectius primordials com a futurs enginyers, adreçat a millorar la futura qualitat de vida d'afectats d'ictus a través d'una nova eina de rehabilitació, més interactiva que ajudarà a suavitzar les dures i llargues sessions a les que es veuen abocats tots els afectats per aquesta malaltia. D'altra banda, haver treballat per primera vegada amb un conjunt de tecnologies apassionants, ha esdevingut un gran esforç i al mateix temps també una gran satisfacció a l'hora de desenvolupar aquest projecte.

Aquest projecte, a més, ha servit com una experiència d'aprenentatge, específicament parlant, altre cop, de la programació amb Unity3D i el llenguatge C# i PHP, que de ben segur servirà per a futurs projectes propis.

Mitjançant aquest treball s'ha pogut aprofundir en la gran utilitat i potencial que poden arribar a tenir les tecnologies de realitat virtual aplicades en camps com el de la fisioteràpia. Tot i així, s'ha observat que resta molt camí per davant, ja que malauradament el preu d'alguns aparells, com per exemple, les ulleres HTC Vive són considerablement elevats per a la assequibilitat que deuria tenir una tecnologia que s'utilitzés per a la tele-rehabilitació. És cert que també queden molts detalls a millorar, com podria ser un exemple l'estabilitat del sistema Kinect. Si bé aquest món era inimaginable fa un temps, qui sap el que pot estar per arribar. Per aquesta raó, es pensa que es seguirà treballant en aquestes tecnologies i, per tant, avançar en aquests camps.

## 11. Línies futures

A partir dels objectius assolits en aquest treball, es poden millorar certs aspectes i modificar/afegir funcionalitats. Alguns d'aquests aspectes són els següents:

- Millorar la interfície, és a dir, millorar l'aspecte dels menús, tant per afavorir que sigui més atractiu com perquè sigui més entenedor.
- Afegir estímuls auditius per part de la fisioterapeuta virtual, és a dir, que la fisioterapeuta doni les pautes per a dur a terme l'exercici i doni indicacions per indicar si el pacient està realitzant correctament l'exercici.
- Afegir una gràfica a temps real; en altres paraules, que les gràfiques que s'obtenen al fitxer Excel es puguin visualitzar mentre es fa l'exercici.
- Millorar el rendiment de l'aplicació. És un paràmetre millorable, ja que a vegades entre escenes triga més del compte o la imatge no es renderitza correctament.
- Fer que els exercicis i jocs siguin graduables, és a dir, que hi hagi certs paràmetres que es puguin graduar, com podria ésser la velocitat d'execució de l'exercici d'abducció horitzontal o el percentatge d'elevació dels polsadors durant el joc de desactivació de la bomba
- Millorar la usabilitat en cas que un pacient li manqui un dels braços o no pugui realitzar els exercicis amb un d'ells, ja que aleshores en el cas d'abducció horitzontal no es comptarien les repeticions, i en el cas del joc de flexió de braços (desactivació la bomba) no es poden desactivar els 2 polsadors.
- Calcular el grau de desviació.
- Millorar o substituir la base de dades. Amb totes les dades que s'exporten en aquest projecte (dades temporals), la base de dades phpMyAdmin queda bastant curta; és per això, que s'hauria d'intentar buscar una altra més eficient i adequada per als valors amb els que es treballen.
- Treballar per afegir jocs i/o exercicis per a les extremitats inferiors.
- Treballar conjunta i contínuament amb fisioterapeutes especialitzats per a analitzar i millorar les dades que s'obtenen, com s'obtenen i altres aspectes de l'aplicació.
- Treballar per tal que l'aplicació sigui realment de tele-rehabilitació. Permetre que el pacient pugui executar l'aplicació i dur a terme els exercicis i jocs sense la necessitat d'un fisioterapeuta proper, és a dir, que pugui realitzar el exercicis a casa sense l'ajuda de ningú.
- Treballar amb pacients reals i comprovar si realment l'aplicació millora el rendiment de rehabilitació i, si és factible, realitzar-se des de casa.

## 12. Pressupost i anàlisi econòmica

En aquest apartat s'especifica, en tot detall, els costs que ha comportat l'elaboració d'aquest projecte. Per fer-ho, s'ha dividit en quatre grans blocs:

### 12.1 Despeses de hardware

El primer bloc consta de les despeses del hardware utilitzat per a que aquest projecte per tal que es pugui desenvolupar correctament. A la Taula 12.1, es mostra el desglossament del cost real de cada aparell durant el temps que s'ha estat treballant.

D'aquesta manera, els costs de hardware total per a aquest projecte és de 186,41€.

Producte	Cost d'adquisició (€)	Cost residual (€)	Vida útil estimada (en mesos)	Depreciació al mes(%)	Temps d'ús (en mesos)	Entitat subministradora	Cost real (€)
Conjunt informàtic del laboratori (ordinador, pantalla, ratolí i teclat)	1500	145,11	48	28,23	4	Visyon360	125,00
Adaptador Kinect/Windows	40	6,85	36	0,92	4	UPC	4,44
Sensor Kinect v2	85	14,55	36	1,96	4	UPC	9,44
Ulleres HTC Vive	499	64,29	42	10,35	4	Visyon360	47,52
<b>Subtotal</b>	<b>2124</b>		-	-	-	-	<b>186,41</b>

Taula 12.1. Costs de hardware



## 12.2 Despeses de software

El segon gran bloc és el de despeses de software. La veritat és que per a la realització d'aquest treball no s'ha hagut d'abonar cap ingrés per cap dels programes, ja que, exceptuant el Unity 3D, es tracta de programes d'*open source* (de llicència gratuïta). Pel que fa al Unity 3D, com que cap dels membres de l'equip té uns ingressos iguals o superiors als 100.000\$ és gratuït.

Així doncs, la taula que es mostra a continuació (Taula 12.2) serveix de resum per recordar els softwares utilitzats i les llicències emprades. D'aquesta manera, els costs de software total per a aquest projecte és de 0€.

Software	Tipus de llicència	Preu (€)
Unity 3D	Unity Personal	0
XAMPP	<i>Open source</i>	0
Microsoft Kinect for Windows SDK	<i>Open source</i>	0
Steam VR	<i>Open source</i>	0
Subtotal	-	0

Taula 12.2. Costs de software

## 12.3 Despeses del consum elèctric

El tercer bloc és el del cost del consum elèctric. Per poder-lo tractar correctament s'ha buscat, per a cada component electrònic, la consumició (en watts) d'energia elèctrica en els seus respectius manuals tècnics. A més, s'ha considerat que la universitat té contractada una tarifa fixa del subministrament elèctric, i l'estat espanyol va fixar pel 2019 un preu de 0.1350€/kWh [42]. També cal tenir present les hores d'ús dels components, que ha estat d'unes 400 hores durant tot el quadrimestre. Així doncs, en la següent taula (Taula 12.3) es mostra el desglossament explicat. D'aquesta manera, el cost del consum elèctric total per a aquest projecte és de 17,28€.

Producte	Consum (W)	Cost (€)
Ordinador	240	12,96
Pantalla	45	2,43
Ratolí	2	0,108
Ulleres HTC Vive	17	0,918
Sensor Kinect v2	16	0,864
<b>Subtotal</b>	<b>320</b>	<b>17,28</b>

Taula 12.3. Cost del consum elèctric

## 12.4 Despeses de personal

L'últim gran bloc de costs és el de personal, és a dir, el sou dels treballadors així com la seva formació inicial. Pel que fa als sous, aquest inclou la seguretat social que si li reté el treballador de la seva nòmina (que és un 6,35% del sou brut[43]), així com també part de la seguretat social de l'empresa (que és d'un 25%).

Fent referència a la formació, es tracta del curs inicial que van dur a terme els enginyers abans de començar a treballar en el projecte.

Així doncs, les següents taules (Taula 12.4 i Taula 12.5) mostren aquesta anàlisi de forma detallada.

D'aquesta manera, el cost de personal total per a aquest projecte és de 36.018€.

Tipus de personal	Hores treballades	Cost de l'hora treballada (€/hora)	Nombre de treballadors	Cost final (€)
Enginyer junior	600	30	2	36.000

Taula 12.4. Sou dels treballadors

Curs	Preu d'adquisició (€)	Nombre de cursos	Cost final (€)
Unity VR Fundamentals	0	2	0
Màster en programació de videojocs amb Unity i C#	9	2	18
<b>Subtotal</b>	-	-	<b>18</b>

Taula 12.5. Formació dels treballadors

## 12.5 Resum de despeses

Finalment, una vegada analitzats els 4 blocs, es presenta una taula resum de tots els costos analitzats per obtenir el cost final del projecte. D'aquesta manera, el cost final del projecte és de 36.221,69€

Tipus de cost	Cost total (€)
Hardware	186,41
Software	0,00
Consum elèctric	17,28
Personal	36.018,00
Total	36.221,69

Taula 12.6. Cost final del projecte

### 13. Bibliografia

- [1] M. C. Howard, "A meta-analysis and systematic literature review of virtual reality rehabilitation programs, review," *Comput. Hum. Behav.*, vol. 70, pp. 317–327, 2017.
- [2] M. L. R.-R. and L. A. I. Olivieri, P. Meriggi, C. Fedeli, E. Brazzoli, A. Castagna, "Computer Assisted REhabilitation (CARE). Lab: A novel approach towards Pediatric Rehabilitation 2.0," *J. Pediatr. Rehabil. Med. An Interdiscip. Approach*, vol. 11, pp. 43–51, 2018.
- [3] M. A. and E. N. A. Borrego, J. Latorre, R. Llorens, "Feasibility of a walking virtual reality system for rehabilitation: objective and subjective parameters," *J. Neuroeng. Rehabil.*, vol. 13, 2016.
- [4] J. Aidaa, "Immersive virtual reality in traumatic brain injury rehabilitation: A literature review," *NeuroRehabilitation*, vol. 42, p. 441/448, 2018.
- [5] M. N. K.-B. S.W. Jerdan, M. Grindle H.C. van Woerden, "Head-Mounted Virtual Reality and Mental Health: Critical Review of Current Research," *JMIR Serious Games*, vol. 6, 2018.
- [6] H. P. and G. C. B. J.E. Deutsch, A.S. Merians, S. Adamovich, "Development and application of virtual reality technology to improve hand use and gait of individuals post- stroke," *Restor Neurol Neurosci*, vol. 22, pp. 371–386, 2004.
- [7] N. . De E.M. Gibbons, A.N. Thomson, "Are virtual reality technologies effective in improving lower limb outcomes for patients following stroke - a systematic review with meta-analysis," *Top Stroke Rehabil.*, vol. 23, pp. 1–18, 2016.
- [8] D. H. P. Y.B. Song, M.H. Chun, W. Kim, S.J. Lee, J.H. Yi, "The effect of virtual reality and tetra-ataxiometric posturography programs on stroke patients with impaired standing balance," *Ann Rehabil Med.*, vol. 3, pp. 160–166, 2014.
- [9] B. H. L. Y.H. Park, C.H. Lee, "Clinical usefulness of the virtual reality- based postural control training on the gait ability in patients with stroke," *J Exerc. Rehabil*, vol. 9, pp. 489–494, 2013.
- [10] H. Y. C. and B. L. C. Yom, "Effects of virtual reality-based ankle exercise on the dynamic balance, muscle tone, and gait of stroke patients," *J Phys Ther Sci*, vol. 27, pp. 845–849, 2015.
- [11] R. K. and M. M. M. D.T. Blake, F. Strata, " , Experience-dependent plasticity in S1 caused by noncoincident inputs," *J Neurophysiol*, vol. 94, pp. 2239–2250, 2005.
- [12] L. Sheehy *et al.*, "Home-based virtual reality training after discharge from hospital-based stroke rehabilitation: A parallel randomized feasibility trial," *Trials*, vol. 20, no. 1, pp. 1–9, 2019, doi: 10.1186/s13063-019-3438-9.
- [13] I. Cikajlo, "Advantages of using 3D virtual reality based training in persons with Parkinson ' s disease : a parallel study," pp. 1–14, 2019.
- [14] M. Gandolfi *et al.*, "Virtual Reality Telerehabilitation for Postural Instability in Parkinson's Disease: A Multicenter, Single-Blind, Randomized, Controlled Trial," *Biomed Res. Int.*, vol. 2017, 2017, doi: 10.1155/2017/7962826.

- [15] Next Level Videogames, "Wii Fit Balance Board (Wii)." [Online]. Available: <http://nextlevelvideogames.com/wii-fit-balance-board-wii/>. [Accessed: 04-Jan-2020].
- [16] J. E. Deutsch, J. A. Lewis, G. Burdea, and S. Member, "Technical and Patient Performance Using a Virtual Reality-Integrated Telerehabilitation System : Preliminary Finding," vol. 15, no. 1, pp. 30–35, 2007.
- [17] Unity, "System requirements for Unity 2019.2," 2019. [Online]. Available: <https://docs.unity3d.com/Manual/system-requirements.html>. [Accessed: 28-Dec-2019].
- [18] Unity, "Aprendiendo la Interfaz," 2016. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/LearningtheInterface.html>. [Accessed: 28-Dec-2019].
- [19] Johel Jiménez Rivera, "C#. Qué es y para que se utiliza," 2018. [Online]. Available: <https://negociosyestrategia.com/blog/que-es-csharp/>. [Accessed: 29-Dec-2019].
- [20] php.net, "¿Qué es PHP?," 2018. [Online]. Available: <https://www.php.net/manual/es/intro-what-is.php>. [Accessed: 30-Dec-2019].
- [21] Carlos Eduardo Plasencia Prado, "¿Qué es y por qué aprender SQL?," 2017. [Online]. Available: <https://devcode.la/blog/que-es-sql/>. [Accessed: 30-Dec-2019].
- [22] A. S. Cueto, *Introducción a phpMyAdmin (parte 1)*. Spain, 2012.
- [23] A. S. Cueto, *Introducción a phpMyAdmin (parte 2)*. Spain, 2012.
- [24] Beatriz Page, "Kinect, un accesorio para jugar que se ha convertido en herramienta para artistas," 28-Apr-2019.
- [25] A. Cahun, "Kinect se retira oficialmente del mercado, Microsoft abandona su fabricación," 2017. [Online]. Available: <https://www.xataka.com.mx/accesorios/kinect-se-retira-oficialmente-del-mercado-microsoft-abandona-su-fabricacion> . [Accessed: 30-Dec-2019].
- [26] J. Romanos, "Kinect consigue un Record Guinness," 2011. [Online]. Available: <http://gizmos.republica.com/videojuegos/kinect-consigue-un-record-guinness.html>. [Accessed: 03-Jan-2020].
- [27] Microsoft, "Configurar Kinect para Windows v2 con un Adaptador de Kinect para un PC con Windows 10," 2019. [Online]. Available: <https://beta.support.xbox.com/help/hardware-network/kinect/kinect-for-windows-v2-setup-with-adapter#32e57597486445be8365fc35eb30d399>. [Accessed: 31-Dec-2019].
- [28] M. S. A. R. D. Collarana, "Plataformas de entretenimiento virtuales usando el sensor Kinect, Unity y técnicas de Gamificación," 2017. [Online]. Available: [http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S1683-07892017000100007](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892017000100007). [Accessed: 31-Dec-2019].
- [29] D. López, "SDK Kinect v2," 2015. [Online]. Available:

- <https://es.slideshare.net/DavidLpez60/kinect-v2-descripcin>. [Accessed: 31-Dec-2019].
- [30] P. A. Vargas, J. J. Guerrero, C. Guevara, and S. S. Gordon, "Technical Contributions to the Quality of Telerehabilitation Platforms: Case Study—ePHoRt Project," 2019. [Online]. Available: [https://www.researchgate.net/figure/Body-points-captured-by-Microsofts-Kinect-6\\_fig1\\_331125193](https://www.researchgate.net/figure/Body-points-captured-by-Microsofts-Kinect-6_fig1_331125193). [Accessed: 19-Dec-2019].
  - [31] A. Murillo, "¿Qué es el SDK para el dispositivo Kinect de Microsoft?," 2012. [Online]. Available: <http://www.kinectfordevelopers.com/es/2012/11/06/que-es-el-sdk-de-microsoft/>. [Accessed: 02-Jan-2020].
  - [32] VIVE, "VIVE," 2018. [Online]. Available: <https://www.vive.com/eu/product/>. [Accessed: 02-Jan-2020].
  - [33] dayofdubai, "HTC Vive Reduces Price To Aed2599 Making The Best Virtual Reality System More Accessible To The Mass Market." [Online]. Available: <https://www.dayofdubai.com/news/htc-vive-reduces-price-aed2599-making-best-virtual-reality-system-more-accessible-mass-market>. [Accessed: 07-Jan-2020].
  - [34] David Pérez, "Así es Steam VR, la apuesta de HTC y Steam por la realidad virtual," 05-Apr-2016.
  - [35] P. de Salud, "Abducción horizontal y aducción horizontal – Anatomía, movimiento, definición." [Online]. Available: <https://periodicosalud.com/abduccion-horizontal-aduccion-horizontal-anatomia-movimiento-definicion/amp/>. [Accessed: 04-Jan-2020].
  - [36] P. Fix, "Shoulder Exercises." [Online]. Available: <https://painfix.co.uk/shoulder/>. [Accessed: 05-Jan-2020].
  - [37] L. Uria, "Planos y ejes corporales." [Online]. Available: <http://anatomiaifisiologia2014.blogspot.com/2014/12/planos-corporales.html>. [Accessed: 04-Jan-2020].
  - [38] Desconegut, "Requerimientos Funcionales y No Funcionales, ejemplos y tips," 2018. [Online]. Available: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>. [Accessed: 22-Dec-2019].
  - [39] L. H. Aguilera, "La basura electrónica y la contaminación ambiental," pp. 46–61, 2010.
  - [40] E. L. P. Europeo, E. L. Consejo, and D. E. L. A. Uni, "L 197/38," pp. 38–71, 2012.
  - [41] Ecotic, "Reciclaje." [Online]. Available: <https://www.ecotic.es/es/246570/Reciclaje.htm>. [Accessed: 10-Jan-2020].
  - [42] Tarifasgasluz, "Precio del kWh: ¿Cuánto vale el kilovatio hora de luz en España?," 2019. [Online]. Available: <https://tarifasgasluz.com/faq/precio-kwh>. [Accessed: 13-Jan-2020].
  - [43] Bankinter, "Cómo calcular tu salario neto," 2019. [Online]. Available: <https://blog.bankinter.com/economia/-/noticia/2017/9/14/como-varia-sueldo-neto-funcion-salario-bruto-graficos>. [Accessed: 13-Jan-2020].



## Annex 1. Instruccions per inicialitzar el joc

### A1.1 Connectar el sensor Kinect

Un cop s'ha engegat l'ordinador, el primer que es pot fer és connectar el sensor Kinect a l'ordinador i al corrent elèctric, amb l'ajuda de l'adaptador Kinect/Windows. Per fer-ho, es seguirà l'esquema de la Figura A1. 1

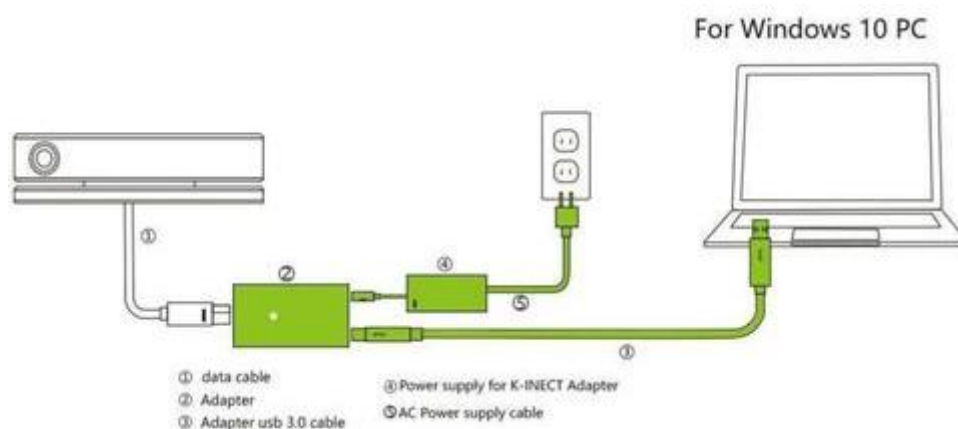


Figura A1. 1 Esquema de connexió del sensor Kinect a l'ordinador i al corrent elèctric [33]

Seguidament, el sensor s'haurà de col·locar entre una alçada mínima de 0.6 metres i una màxima de 1.8 metres i l'usuari que el vagi a utilitzar haurà de romandre davant seu a una distància mínima de 1.4 metres (veure la Figura A1. 2).

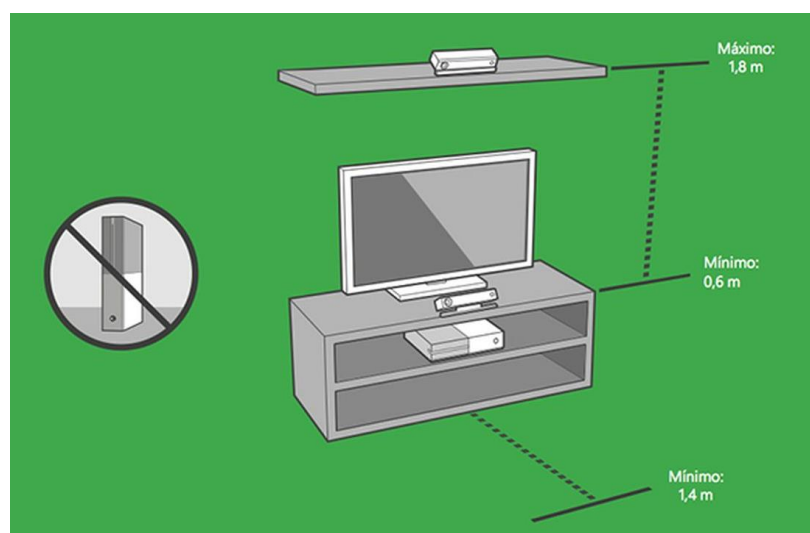


Figura A1. 2. Col·locació del sensor Kinect [34]



## A1.2 Connectar les ulleres HTC Vive

Per connectar les ulleres de RV a l'ordinador, així com tot el hardware que el comporta, la pròpia pàgina de Vive ofereix unes instruccions detallades, que són acompanyades d'unes fotografies i d'un vídeo explicatiu per a que l'usuari no tingui cap mena de dubte del procediment a seguir. El següent enllaç, [https://www.vive.com/mx/support/vive/category\\_howto/connecting-the-headset-to-your-computer.html](https://www.vive.com/mx/support/vive/category_howto/connecting-the-headset-to-your-computer.html), connecta amb la pàgina oficial de Vive, just a l'apartat de les instruccions esmentades anteriorment.

## A1.3 Iniciar la base de dades

Per iniciar la base de dades, primerament s'ha d'activar els mòduls d'"Apache" i "MySQL" des del XAMPP, clicant sobre el botó "Start" de la columna "Actions" de cada mòdul en particular.

Es sap que s'han iniciat correctament quan apareixen els seus noms en el menú principal del XAMPP marcats en verd (veure Figura A1. 3).

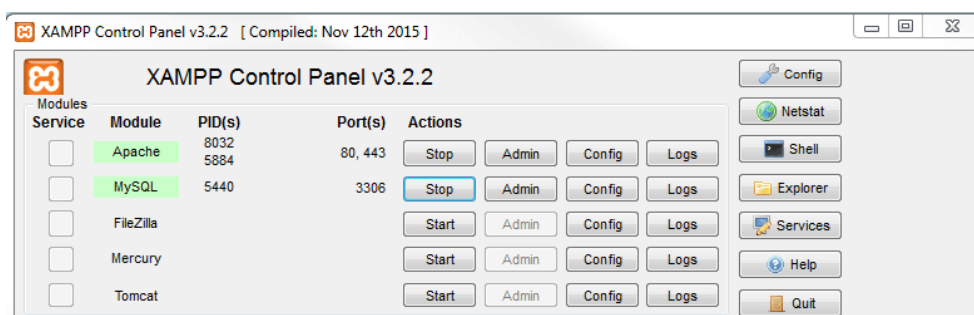


Figura A1. 3 Activació dels mòduls Apache i MySQL

## Annex 2. Adjunt de l'aplicació

En aquest apartat de l'annex s'adjunta l'enllaç on està penjat tot el projecte. S'hi pot trobar la pròpia aplicació desenvolupada amb el motor de videojocs Unity, juntament amb els scripts corresponents (aquests mateixos es poden veure a l'annex 3 d'aquesta mateixa memòria), així com també els fitxers Excel que contenen totes les dades obtingudes en les quatre proves que es van realitzar.

Enllaç: [https://drive.google.com/open?id=1nX\\_MBBDUazNoxjjSzywlpJdQ7fl6KVbf](https://drive.google.com/open?id=1nX_MBBDUazNoxjjSzywlpJdQ7fl6KVbf)

## Annex 3. Codis

### A3.1 Codis C#

#### Actions

```

1  using UnityEngine;
2  using System.Collections;
3
4  [RequireComponent (typeof (Animator))]
5  public class Actions : MonoBehaviour {
6
7      private Animator animator;
8      public bool flex = false;
9      public bool ABDUP = false;
10     public bool HABDUP = false;
11     public bool idle = false;
12
13     // const int countOfDamageAnimations = 3;
14     //int lastDamageAnimation = -1;
15
16     void Awake () {
17         animator = GetComponent<Animator> ();
18     }
19
20     public void Update()
21     {
22         if (flex == true)
23         {
24             //animator.SetBool("PosInicial", true);
25             animator.SetBool("Flex", false);
26             animator.SetBool("ABDUP", false);
27             animator.SetBool("HABDUP", false);
28             StartCoroutine(FlexWait());
29         }
30
31         if (ABDUP == true)
32         {
33             animator.SetBool("ABDUP", false);
34             animator.SetBool("Flex", false);
35             animator.SetBool("HABDUP", false);
36             StartCoroutine(ABDUPWait());
37         }
38
39         if (HABDUP == true)
40         {
41             animator.SetBool("HABDUP", false);
42             animator.SetBool("Flex", false);
43             animator.SetBool("ABDUP", false);
44             StartCoroutine(HABDUPWait());
45         }
46     }
47
48     public void Flex()
49     {
50         //flex = !flex;
51         HABDUP = false;
52         flex = true;
53         ABDUP = false;
54     }
55
56     public void ABDUPP()

```

```

57     {
58         HABDUP = false;
59         ABDUP = true;
60         flex = false;
61     }
62
63     public void HBDUPP()
64     {
65         HBDUP = true;
66         ABDUP = false;
67         flex = false; 68
68     }
69     public void idles()
70     {
71         idle = true;
72         flex = false;
73         ABDUP = false;
74         HBDUP = false;
75         animator.SetBool("Idle", true);
76         animator.SetBool("Flex", false);
77         animator.SetBool("ABDUP", false);
78         animator.SetBool("HBDUP", false); 79
79     }
80
81
82     IEnumerator FlexWait()
83     {
84         yield return new WaitForSeconds(3);
85         //animator.SetBool("PosInicial", false);
86         animator.SetBool("Flex", true); 87
87     }
88     IEnumerator ABDUPWait()
89     {
90         yield return new WaitForSeconds(3);
91         //animator.SetBool("PosInicial", false);
92         animator.SetBool("ABDUP", true); 93
93     }
94
95     IEnumerator HBDUPWait()
96     {
97         yield return new WaitForSeconds(3);
98         //animator.SetBool("PosInicial", false);
99         animator.SetBool("HBDUP", true);
100     }
101
102     /*public void Stay () {
103         animator.SetBool("Aiming", false);
104         animator.SetFloat ("Speed", 0f);
105     }
106
107     public void Walk () {
108         animator.SetBool("Aiming", false);
109         animator.SetFloat ("Speed", 0.5f);
110     }
111
112     public void Run () {

```

```
113     animator.SetBool("Aiming", false);
114     animator.SetFloat ("Speed", 1f);
115 }
116
117 public void Attack () {
118     Aiming ();
119     animator.SetTrigger ("Attack");
120 }
121
122 public void Death () {
123     if (animator.GetCurrentAnimatorStateInfo (0).IsName ("Death"))
124         animator.Play("Idle", 0);
125     else
126         animator.SetTrigger ("Death");
127 }
128
129 public void Damage () {
130     if (animator.GetCurrentAnimatorStateInfo (0).IsName ("Death")) return;
131     int id = Random.Range(0, countOfDamageAnimations);
132     if (countOfDamageAnimations > 1)
133         while (id == lastDamageAnimation)
134             id = Random.Range(0, countOfDamageAnimations);
135     lastDamageAnimation = id;
136     animator.SetInteger ("DamageID", id);
137     animator.SetTrigger ("Damage");
138 }
139
140 public void Jump () {
141     animator.SetBool ("Squat", false);
142     animator.SetFloat ("Speed", 0f);
143     animator.SetBool("Aiming", false);
144     animator.SetTrigger ("Jump");
145 }
146
147 public void Aiming () {
148     animator.SetBool ("Squat", false);
149     animator.SetFloat ("Speed", 0f);
150     animator.SetBool("Aiming", true);
151 }
152
153 public void Sitting () {
154     animator.SetBool ("Squat", !animator.GetBool("Squat"));
155     animator.SetBool("Aiming", false);
156 }
157 */
158 }
159
```

## Angulos

```

1  using System.Collections;
2  using UnityEngine;
3  using UnityEngine.UI;
4  using UnityEngine.SceneManagement;
5  using System.Collections.Generic; //
6  using System;
7  using System.Text;
8  using UnityEngine.EventSystems;
9  using System.Text.RegularExpressions;
10 using System.Linq;
11
12
13 /// <summary>
14 ///
15 ///
16 /// </summary>
17 ///
18
19 public class Angulos : MonoBehaviour {
20
21     public GameObject DDNI;
22     private Text DNI;
23     private string DNIII;
24     public Text Content;
25     public string paciente;
26     public string Dia;
27
28     public GameObject Fechaa;
29     private Text Fecha;
30
31     string LoginURL = "localhost/users/listangulos.php";
32
33     // Use this for initialization
34     void Start()
35     {
36         DNI = DDNI.GetComponent<Text>();//Nombre Paciente
37         Fecha = Fechaa.GetComponent<Text>(); 38
38     }
39
40     void Update()
41     {
42         string paciente = DNI.text;
43         string paaciente = paciente.Remove(0, 1);
44
45         string Dia = Fecha.text;
46         string Diaa = Dia.Remove(0, 1);
47         StartCoroutine(Lista1(paaciente, Diaa)); 48
48     }
49
50     IEnumerator Lista1(string paaciente, string Diaa)
51     {
52         WWWForm form = new WWWForm();
53         form.AddField("PatientNamePost", paaciente);//
54         form.AddField("DiaPost", Diaa);
55         Debug.Log(Diaa);
56     }

```

```
57         Debug.Log(paaciente);
58
59         WWW www = new WWW(LoginURL, form);
60
61         yield return www;
62
63         Content.text = www.text;
64     }
65 }
66
67
```

**BodySourceManager**

```

1  using UnityEngine;
2  using System.Collections;
3  using Windows.Kinect;
5  public class BodySourceManager : MonoBehaviour
6  {
7      private KinectSensor _Sensor;
8      private BodyFrameReader _Reader;
9      private Body[] _Data = null;
10
11
12     public Body[] GetData()
13     {
14         return _Data;
15     }
16
17     public Windows.Kinect.Vector4 FloorPlanes
18     {
19         get;
20         private set; 21
21     }
22
23     void Start ()
24     {
25         _Sensor = KinectSensor.GetDefault();
26
27         if (_Sensor != null)
28         {
29             _Reader = _Sensor.BodyFrameSource.OpenReader();
30
31             if (!_Sensor.IsOpen)
32             {
33                 _Sensor.Open();
34             }
35         }
36     }
37
38     void Update ()
39     {
40         if (_Reader != null)
41         {
42             var frame = _Reader.AcquireLatestFrame();
43             if (frame != null)
44             {
45                 if (_Data == null)
46                 {
47                     _Data = new Body[_Sensor.BodyFrameSource.BodyCount];
48                 }
49
50                 frame.GetAndRefreshBodyData(_Data);
51                 frame.Dispose();
52                 frame = null;
53             }
54         }
55     }
56 }

```



```
57
58     void OnApplicationQuit()
59     {
60         if (_Reader != null)
61         {
62             _Reader.Dispose();
63             _Reader = null;
64         }
65
66         if (_Sensor != null)
67         {
68             if (_Sensor.IsOpen)
69             {
70                 _Sensor.Close();
71             }
72
73             _Sensor = null;
74         }
75     }
76 }
77
78
79
80
```

**BodySourceUpperBody**

```

1  using UnityEngine;
2  using System.Collections;
3  using Windows.Data;
4  using System.Collections.Generic;
5  using Kinect = Windows.Kinect;
6  using Windows.Kinect;
7  using System.Linq;
8  using System.IO;
9  using System;
10 using UnityEngine.UI;
11 using System.Threading;
12 using UnityEngine.SceneManagement;
13
14 public class BodySourceUpperBody : MonoBehaviour
15 {
16     public Material BoneMaterial;
17     public GameObject BodySourceManager;
18     public GameObject _personaje1_Unity;
19     public GameObject Master;
20     public GameObject Reference;
21     public GameObject Hips;
22     public GameObject LeftUpLeg;
23     public GameObject LeftLeg;
24     public GameObject RightUpLeg;
25     public GameObject RightLeg;
26     public GameObject Spine;
27     public GameObject Spine1;
28     public GameObject Spine2;
29     public GameObject LeftShoulder;
30     public GameObject LeftArm;
31     public GameObject LeftForeArm;
32     public GameObject LeftHand;
33     public GameObject Neck;
34     public GameObject Head;
35     public GameObject RightShoulder;
36     public GameObject RightArm;
37     public GameObject RightForeArm;
38     public GameObject RightHand;
39     public GameObject LeftFoot; //
40     public GameObject RightFoot; //
41     public GameObject RightAnkle; //
42     public GameObject LeftAnkle; //
43     public float a = 0;
44     public float a2 = 0;
45     public float b = 0;
46     public float b2 = 0;
47     public float maxtot = 0;
48     public Text derecha = null;
49     public Text izquierda = null;
50     public int cont = -1;
51     public int conta = 0;
52     public bool rep = true;
53     public Text angulo = null;
54     public Text angulo2 = null;

```

```

57     public bool angulonum = false;
58     public GameObject semaforo;
59     public GameObject semaforo2;
60     public Text copiaej = null;
61     public bool Elbow=false;
62     public bool ABDUP = false;
63     public bool HABDUP = false;
64     public float valor;
65     public float valor2;
66     public Text Simetria;
67     private Dictionary<ulong, GameObject> _Bodies = new Dictionary<ulong,      ↗
        GameObject>();
68     private BodySourceManager _BodyManager;
69     public Text Simetrianum;
70     public Text repeticionesnum;
71     public GameObject repeticionesHABDUP;
72     public bool repeticiones0=false;
73     public float[] ang;
74     public Text AngleR;
75     public Text AngleL;
76     public GameObject[] DNI;
77     public string nombre;
78     public float angledreta;
79     private string dataDia;
80
81     public static float HABDUptime;
82     public static string repHABDUP;
83     public GameObject semaforoHABDUP;
84     public GameObject semaforo2HABDUP;
85     //public Text guardar2;
86
87
88
89     /// <summary>
90     /// PARTE CANVAS MAX VALUE
91     /// </summary>
92     public bool flex = false;
93
94     public void Flex()
95     {
96         flex = true;
97         StartCoroutine(Seemaforo()); 98}
99
100    public void Elboww() {
101        Elbow = !Elbow;
102        StartCoroutine(Seemaforo());
103    }
104
105    public void ABDUPP()
106    {
107        ABDUP = true;
108        StartCoroutine(Seemaforo());
109    }
110
111    public void HABDUPP()

```

```

112     {
113         HABDUP = true;
114         StartCoroutine(SeemaforoHABDUP());
115         HABDUPtime = 0;
116
117     }
118 }
119
120 public void excelHABDUP(float HABDUPright, float HABDUPleft, float
    FLEXright, float FLEXleft, Vector3 shoulderright, Vector3 shoulderleft,
    Vector3 wristright, Vector3 wristleft, Vector3 spinemidv, Vector3
    Elbowright, Vector3 Elbowleft, Vector3 SpineB, Vector3 neck, Vector3
    head, float HABDUPtime)
121 {
122     string path5 = Application.dataPath + "/Seguimiento/"+nombre+"/
        HAbduction" + dataDia + ".xls"; //Creamos un documento de excel
        (.xls) para guardar los datos
123     string content5 = HABDUPtime+"\t"
124         +shoulderright.x+"\t" + shoulderright.y + "\t" + shoulderright.z
        + "\t"
125         + shoulderleft.x+ "\t"+shoulderleft.y + "\t" + shoulderleft.z+
        "\t"
126         + Elbowright.x + "\t" + Elbowright.y + "\t" + Elbowright.z + "\t"
127         + Elbowleft.x + "\t" + Elbowleft.y + "\t" + Elbowleft.z + "\t"
128         + wristright.x + "\t"+ wristright.y + "\t" + wristright.z + "\t"
129         + wristleft.x + "\t" + wristleft.y + "\t" + wristleft.z + "\t"
130         + spinemidv.x + "\t" + spinemidv.y + "\t" + spinemidv.z + "\t"
131         + SpineB.x + "\t" + SpineB.y + "\t" + SpineB.z + "\t"
132         + neck.x + "\t" + neck.y + "\t" + neck.z + "\t"
133         + head.x + "\t" + head.y + "\t" + head.z + "\t"
134         + HABDUPright + "\t" + HABDUPleft + "\t"
135         + FLEXright + "\t" + FLEXleft + "\n"; //Contenido del .xls
136
137     if (!File.Exists(path5))
138     {
139         File.AppendAllText(path5, "Data Angulo Derecho\nLogin date: " +
            System.DateTime.Now + "\n" +
140             "Tiempo \t"+
141             "Hombro Derecho X \t Hombro Derecho Y \t Hombro Derecho Z \t"
            +
142             "Hombro Izquierdo X \t Hombro Izquierdo Y \t Hombro Izquierdo
            Z \t" +
143             "Codo Derecho X \t Codo Derecho Y \t Codo Derecho Z \t" +
144             "Codo Izquierdo X \t Codo Izquierdo Y \t Codo Izquierdo Z \t"
            +
145             "Muneca Derecha X \t Muneca Derecha Y \t Muneca Derecha Z\t"
            +
146             "Muneca Izquierda X \t Muneca Izquierda Y \t Muneca Izquierda
            Z \t" +
147             "Spine Mid X \t Spine Mid Y \t Spine Mid Z \t" +
148             "Spine Base X \t Spine Base Y \t Spine Base Z \t" +
149             "Cuello X \t Cuello Y \t Cuello Z \t" +
150             "Cabeza X \t Cabeza Y \t Cabeza Z\t"+
151             "Angulo HABDUP derecho \tAngulo HABDUP izquierdo \t"+
152             "Angulo FLEX derecho \t Angulo FLEX izquierdo\n");
153     }

```

```

154         File.AppendAllText(path5, content5);
155     }
156
157     /*public void sceneBomba()
158     {
159
160         SceneManager.LoadScene("Joc Bomba");
161     }*/
162
163
164
165
166
167
168     /// <summary>
169     /// //////////////////////////////////////
170     /// </summary>
171
172     // Mapeo "puntos" del Cuerpo Humano
173
174     private Dictionary<Kinect.JointType, Kinect.JointType> _BoneMap = new Dictionary<Kinect.JointType, Kinect.JointType>()
175     {
176         { Kinect.JointType.FootLeft, Kinect.JointType.AnkleLeft },
177         { Kinect.JointType.AnkleLeft, Kinect.JointType.KneeLeft },
178         { Kinect.JointType.KneeLeft, Kinect.JointType.HipLeft },
179         { Kinect.JointType.HipLeft, Kinect.JointType.SpineBase },
180
181         { Kinect.JointType.FootRight, Kinect.JointType.AnkleRight },
182         { Kinect.JointType.AnkleRight, Kinect.JointType.KneeRight },
183         { Kinect.JointType.KneeRight, Kinect.JointType.HipRight },
184         { Kinect.JointType.HipRight, Kinect.JointType.SpineBase },
185
186         { Kinect.JointType.HandTipLeft, Kinect.JointType.HandLeft },
187         { Kinect.JointType.ThumbLeft, Kinect.JointType.HandLeft },
188         { Kinect.JointType.HandLeft, Kinect.JointType.WristLeft },
189         { Kinect.JointType.WristLeft, Kinect.JointType.ElbowLeft },
190         { Kinect.JointType.ElbowLeft, Kinect.JointType.ShoulderLeft },
191         { Kinect.JointType.ShoulderLeft, Kinect.JointType.SpineShoulder },
192
193         { Kinect.JointType.HandTipRight, Kinect.JointType.HandRight },
194         { Kinect.JointType.ThumbRight, Kinect.JointType.HandRight },
195         { Kinect.JointType.HandRight, Kinect.JointType.WristRight },
196         { Kinect.JointType.WristRight, Kinect.JointType.ElbowRight },
197         { Kinect.JointType.ElbowRight, Kinect.JointType.ShoulderRight },
198         { Kinect.JointType.ShoulderRight, Kinect.JointType.SpineShoulder },
199
200         { Kinect.JointType.SpineBase, Kinect.JointType.SpineMid },
201         { Kinect.JointType.SpineMid, Kinect.JointType.SpineShoulder },
202         { Kinect.JointType.SpineShoulder, Kinect.JointType.Neck },
203         { Kinect.JointType.Neck, Kinect.JointType.Head },
204     };
205
206     public void Start()
207     {
208

```

```

209     _personaje1_Unity = GameObject.Find("personaje1_Unity");
210     Master = GameObject.Find("master");
211     Reference = GameObject.Find("Reference");
212     Hips = GameObject.Find("Hips");
213     LeftUpLeg = GameObject.Find("LeftUpLeg");
214     LeftLeg = GameObject.Find("LeftLeg");
215     RightUpLeg = GameObject.Find("RightUpLeg");
216     RightLeg = GameObject.Find("RightLeg");
217     Spine = GameObject.Find("Spine");
218     Spine1 = GameObject.Find("Spine1");
219     Spine2 = GameObject.Find("Spine2");
220     LeftShoulder = GameObject.Find("LeftShoulder");
221     LeftArm = GameObject.Find("LeftArm");
222     LeftForeArm = GameObject.Find("LeftForeArm");
223     LeftHand = GameObject.Find("LeftHand");
224     Neck = GameObject.Find("Neck");
225     Head = GameObject.Find("Head");
226     RightShoulder = GameObject.Find("RightShoulder");
227     RightArm = GameObject.Find("RightArm");
228     RightForeArm = GameObject.Find("RightForeArm");
229     RightHand = GameObject.Find("RightHand");
230     RightAnkle = GameObject.Find("RightToeBase");
231     LeftAnkle = GameObject.Find("LeftToeBase");
232     RightFoot = GameObject.Find("RightFoot");
233     LeftFoot = GameObject.Find("LeftFoot");
234
235
236
237
238     #region Nova entrada en els.xls
239
240     DNI = GameObject.FindGameObjectsWithTag("DNI");
241     nombre = DNI[1].name;
242     dataDia = System.DateTime.Now.ToString();
243     dataDia = dataDia.Replace('/', '_');
244     dataDia = dataDia.Replace(':', '.');
245
246     if (!Directory.Exists(Application.dataPath + "/Seguimiento/" + nombre ))
247     {
248         Directory.CreateDirectory(Application.dataPath + "/Seguimiento/" + nombre);
249     }
250
251     string path = Application.dataPath + "/Seguimiento/"+nombre+"/Abduction_AngleR_" + dataDia + ".xls";
252
253     if (File.Exists(path))
254     {
255         File.AppendAllText(path, "\nLogin date: " + System.DateTime.Now + "\n");
256     }
257
258
259     string path2 = Application.dataPath + "/Seguimiento/"+nombre+"/Abduction_AngleL_" + dataDia + ".xls";

```

```

260
261     if (File.Exists(path2))
262     {
263         File.AppendAllText(path2, "\nLogin date: " + System.DateTime.Now
264             + "\n");
265     }
266     string path3 = Application.dataPath + "/Seguimiento/" + nombre + "/"
267         Flex_AngleR_" + dataDia + ".xls";
268     if (File.Exists(path3))
269     {
270         File.AppendAllText(path3, "\nLogin date: " + System.DateTime.Now
271             + "\n");
272     }
273
274     string path4 = Application.dataPath + "/Seguimiento/" + nombre + "/"
275         Flex_AngleL_" + dataDia + ".xls";
276     if (File.Exists(path4))
277     {
278         File.AppendAllText(path4, "\nLogin date: " + System.DateTime.Now
279             + "\n");
280     }
281     string path5 = Application.dataPath + "/Seguimiento/" + nombre + "/"
282         HAbduction_AngleR_" + dataDia + ".xls";
283     if (File.Exists(path5))
284     {
285         File.AppendAllText(path5, "\nLogin date: " + System.DateTime.Now
286             + "\n");
287     }
288
289     string path6 = Application.dataPath + "/Seguimiento/" + nombre + "/"
290         HAbduction_AngleL_" + dataDia + ".xls";
291     if (File.Exists(path6))
292     {
293         File.AppendAllText(path6, "\nLogin date: " + System.DateTime.Now
294             + "\n");
295     }
296     #endregion
297
298
299
300
301     IEnumerator Seemaforo()
302     {
303         copiaej.text = "Copie la posicion de la entrenadora!";
304         yield return new WaitForSeconds(3);
305         Destroy(semaforo);
306         copiaej.text = "Ahora es su turno!";

```

```

307     semaforo2.SetActive(true);
308 }
309
310 IEnumerator SeemaforoHABDUP()
311 {
312     yield return new WaitForSeconds(3);
313
314     semaforoHABDUP.SetActive(false);
315
316     semaforo2HABDUP.SetActive(true);
317 }
318
319
320
321 IEnumerator Temporizador(float angleFlex, float angleFlex1)
322 {
323     flex = false;
324     yield return new WaitForSeconds(0.5f); //Mira angulos FLex Cada ↗
325         segundo
326     MAX(angleFlex, angleFlex1);
327     flex = true;
328 }
329 IEnumerator TemporizadorABDUP(float correct, float correct1)
330 {
331
332     ABDUP = false;
333     yield return new WaitForSeconds(0.5f); //Mira angulos FLex Cada ↗
334         segundo
335     MAXABDUP(correct, correct1);
336     ABDUP = true;
337 }
338
339 IEnumerator TemporizadorHABDUP(float HABDUPleft, float HABDUPright, float ↗
340     FLEXleft, float FLEXright, Vector3 shoulderright, Vector3 shoulderleft, ↗
341     Vector3 wristright, Vector3 wristleft, Vector3 spinemidv, Vector3 ↗
342     Elbowright, Vector3 Elbowleft, Vector3 SpineB, Vector3 neck, Vector3 ↗
343     head, float HABDUPtime)
344 {
345     HABDUP = false;
346     yield return new WaitForSeconds(0.2f);
347
348     if (HABDUPright > a)
349     {
350         a = HABDUPright;
351     }
352
353     if (HABDUPleft > a2)
354     {
355         a2 = HABDUPleft;
356     }
357
358     excelHABDUP(HABDUPright, HABDUPleft, FLEXright, FLEXleft, ↗
359         shoulderright, shoulderleft, wristright, wristleft, spinemidv, ↗

```



```

        Elbowright, Elbowleft, SpineB, neck, head, HABDUPtime);
356
357     HABDUP = true;
358
359 }
360 IEnumerator TemporizadorBOMBA(float HABDUPleft, float HABDUPright, float
    FLEXleft, float FLEXright, Vector3 shoulderright, Vector3 shoulderleft,
    Vector3 wristright, Vector3 wristleft, Vector3 spinemidv, Vector3
    Elbowright, Vector3 Elbowleft, Vector3 SpineB, Vector3 neck, Vector3
    head)
361 {
362     Debug.Log("enumerator");
363     if (bomba.PlayBomb == 1)
364     {
365         bomba.PlayBomb = 0;
366         yield return new WaitForSeconds(0.2f); //Mira angulos FLEX Cada
            segundo
367         Debug.Log("playbomb=1");
368
369
370         string path6 = Application.dataPath + "/Seguimiento/" + nombre +
            "/Bomba" + dataDia + ".xls"; //Creamos un documento de excel
            (.xls) para guardar los datos
371         string content6 = bomba.Bombtime + "\t"
372             + shoulderright.x + "\t" + shoulderright.y + "\t" +
            shoulderright.z + "\t"
373             + shoulderleft.x + "\t" + shoulderleft.y + "\t" +
            shoulderleft.z + "\t"
374             + Elbowright.x + "\t" + Elbowright.y + "\t" + Elbowright.z +
            "\t"
375             + Elbowleft.x + "\t" + Elbowleft.y + "\t" + Elbowleft.z +
            "\t"
376             + wristright.x + "\t" + wristright.y + "\t" + wristright.z +
            "\t"
377             + wristleft.x + "\t" + wristleft.y + "\t" + wristleft.z +
            "\t"
378             + spinemidv.x + "\t" + spinemidv.y + "\t" + spinemidv.z +
            "\t"
379             + SpineB.x + "\t" + SpineB.y + "\t" + SpineB.z + "\t"
380             + neck.x + "\t" + neck.y + "\t" + neck.z + "\t"
381             + head.x + "\t" + head.y + "\t" + head.z + "\t"
382             + HABDUPright + "\t" + HABDUPleft + "\t"
383             + FLEXright + "\t" + FLEXleft + "\n"; //Contenido del .xls
384
385         if (!File.Exists(path6))
386         {
387             Debug.Log("file exists");
388
389             File.AppendAllText(path6, "Data Angulo Derecho\nLogin date: "
                + System.DateTime.Now + "\n" +
390                 "Tiempo \t" +
391                 "Hombro Derecho X \t Hombro Derecho Y \t Hombro Derecho Z
                    \t" +
392                 "Hombro Izquierdo X \t Hombro Izquierdo Y \t Hombro
                    Izquierdo Z \t" +
393                 "Codo Derecho X \t Codo Derecho Y \t Codo Derecho Z \t" +

```

```

394         "Codo Izquierdo X \t Codo Izquierdo Y \t Codo Izquierdo Z \t" +
395         "Muneca Derecha X \t Muneca Derecha Y \t Muneca Derecha Z \t" +
396         "Muneca Izquierda X \t Muneca Izquierda Y \t Muneca Izquierda Z \t" +
397         "Spine Mid X \t Spine Mid Y \t Spine Mid Z \t" +
398         "Spine Base X \t Spine Base Y \t Spine Base Z \t" +
399         "Cuello X \t Cuello Y \t Cuello Z \t" +
400         "Cabeza X \t Cabeza Y \t Cabeza Z \t" +
401         "Angulo HABDUP derecho \t Angulo HABDUP izquierdo \t" +
402         "Angulo FLEX derecho \t Angulo FLEX izquierdo\n");
403     }
404
405     File.AppendAllText(path6, content6);
406     bomba.PlayBomb = 1;
407
408 }
409 }
410
411
412
413
414
415 public void MAXABDUP(float correct, float correct1)
416 {
417
418     if (correct > a)
419     {
420         a = correct;
421
422     }
423
424     if (correct1 > a2)
425     {
426         a2 = correct1;
427     }
428     valor = correct; // derecha
429
430     valor2 = correct1; // izquierda
431
432     #region Abduction_AngleR_
433
434     // AngleR.text = ("" + valor); // Creamos el vector para los angulos
435     // (frame a frame) del brazo derecho
436     // Debug.Log(AngleR.text)
437     // AngleL.text = ("" + valor2); // Creamos el vector para los
438     // angulos (frame a frame) del brazo izquierdo
439
440     string path = Application.dataPath + "/Seguimiento/" + nombre + "/" +
441     "Abduction_AngleR_" + dataDia + ".xls"; //Creamos un documento de
442     excel (.xls) para guardar los datos
443     string content = valor + "\n"; //Contenido del .xls
444
445     if (!File.Exists(path))
446     {

```

```

443         File.AppendAllText(path, "Data Angulo Derecho\nLogin date: " +
444             System.DateTime.Now + "\n");
445     }
446     File.AppendAllText(path, content);
447     #endregion
448
449     #region Abduction_AngleL_
450     string path2 = Application.dataPath + "/Seguimiento/" + nombre + "/"
451         Abduction_AngleL_" + dataDia + ".xls"; //Creamos un documento de
452         excel (.xls) para guardar los datos
453     string content2 = valor2 + "\n"; //Contenido del .xls
454
455     if (!File.Exists(path2))
456     {
457         File.WriteAllText(path2, "Data Angulo Izquierdo\nLogin date: " +
458             System.DateTime.Now + "\n");
459     }
460
461     File.AppendAllText(path2, content2);
462     #endregion
463
464     //guardar = valor + valor2;
465
466     // guardar2.text = ( guardar2.text + valor + "/" + valor2 + ";");
467
468     Repeticiones(valor);
469     derecha.text = ("" + correct); //Actual
470     izquierda.text = ("" + correctl); //Actualiz
471     angulo.text = ("" + a); //Valor maximo derecho
472     angulo2.text = ("" + a2); //Valor maximo iz
473     if (((correct - correctl) > b))
474     {
475         b = correct - correctl;
476         Simetria.text = "Esta desviado la derecha en : " + b + " Grados";
477     }
478
479     if (((correctl - correct) > b))
480     {
481         b = correctl - correct;
482         Simetria.text = "Esta desviado la izquierda en : " + b + "
483             Grados";
484         Simetrianum.text = ("" + b);
485         //Debug.Log(b);
486     }
487
488
489
490 }
491
492
493 public void MAX(float angleFlex, float angleFlexl)

```

```

494     {
495
496
497
498         if (angleFlex > a)
499         {
500             a = angleFlex;
501
502         }
503         if (angleFlex1 > a2)
504         {
505             a2 = angleFlex1;
506
507         }
508         valor = angleFlex;
509         valor2 = angleFlex1;
510
511         //AngleR.text = ("" + valor); // Creamos el vector para los angulos
512         //Debug.Log(AngleR.text)
513         //AngleL.text = ("" + valor2); // Creamos el vector para los angulos
514         //Debug.Log(AngleL.text)
515
516         #region Flex_AngleR_
517         string path3 = Application.dataPath + "/Seguimiento"+nombre+"/
518             Flex_AngleR_" + dataDia + ".xls"; //Creamos un documento de excel
519             (.xls) para guardar los datos
520         string content3 = valor + "\n"; //Contenido del .xls
521
522         if (!File.Exists(path3))
523         {
524             File.AppendAllText(path3, "Data Angulo Derecho\nLogin date: " +
525                 System.DateTime.Now + "\n");
526         }
527
528         File.AppendAllText(path3, content3);
529
530         #endregion
531
532         #region Flex_AngleL_
533         string path4 = Application.dataPath + "/Seguimiento" + nombre + "/"
534             Flex_AngleL_" + dataDia + ".xls"; //Creamos un documento de excel
535             (.xls) para guardar los datos
536         string content4 = valor2 + "\n"; //Contenido del .xls
537
538         if (!File.Exists(path4))
539         {
540             File.AppendAllText(path4, "Data Angulo Izquierdo\nLogin date: " +
541                 System.DateTime.Now + "\n");
542         }
543
544         File.AppendAllText(path4, content4);
545
546         #endregion
547
548     }
549
550 }

```

```

542
543     Repeticiones(valor);
544     derecha.text = ("" + angleFlex);
545     izquierda.text = ("" + angleFlex1);
546     angulo.text = (""+ a); //si falla al hacer graficas, poner un "0".
547     angulo2.text = ("" + a2);
548     //PONER LAS REPETIVIONES-> CONT
549     if (((angleFlex - angleFlex1) > b))
550     {
551         b = angleFlex - angleFlex1;
552         Simetria.text = "Esta desviado la derecha en :" + b + " Grados";
553         Simetrianum.text = ("" + b);
554
555     }
556     if (((angleFlex1 - angleFlex) > b))
557     {
558         b = angleFlex1 - angleFlex;
559         Simetria.text = "Esta desviado la izquierda en :" + b + "
Grados";
560
561     }
562 }
563
564 public void Repeticiones(float valor)
565 {
566     repeticionesnum.text = ("" + cont);
567     if (valor < 20 && rep == true)
568     {
569         if (repeticiones0 == true)
570         {
571             cont = cont + 1;
572
573         }
574
575         rep = false;
576         Debug.Log(cont);
577
578
579         if (cont == 5) //Mirar poner un 5
580         {
581             Debug.Log("Valor maximo enviado : " + a);
582
583         }
584
585     }
586     if (valor > 20)
587     {
588         rep = true;
589         repeticiones0 = true;
590
591     }
592
593
594 }
595 void Update()
596 {

```

```

597     Debug.Log(bomba.PlayBomb);
598     HABDUPTime += Time.deltaTime;
599
600     if (BodySourceManager == null)
601     {
602         return;
603     }
604
605     _BodyManager = BodySourceManager.GetComponent<BodySourceManager>();
606     if (_BodyManager == null)
607     {
608         return;
609     }
610
611     Kinect.Body[] data = _BodyManager.GetData();
612     if (data == null)
613     {
614         return;
615     }
616
617     var floorPlane = _BodyManager.FloorPlanes;
618     var newrotation = Quaternion.FromToRotation(new Vector3(floorPlane.X, ↗
        floorPlane.Y, floorPlane.Z), Vector3.up);
619
620
621     List<ulong> trackedIds = new List<ulong>();
622     foreach (var body in data)
623     {
624         if (body == null)
625         {
626             continue;
627         }
628
629         if (body.IsTracked)
630         {
631             trackedIds.Add(body.TrackingId);
632         }
633     }
634
635     List<ulong> knownIds = newList<ulong>(_Bodies.Keys);
636
637     // First delete untracked bodies
638     foreach (ulong trackingId in knownIds)
639     {
640         if (!trackedIds.Contains(trackingId))
641         {
642
643             Destroy(_Bodies[trackingId]);
644             _Bodies.Remove(trackingId);
645         }
646     }
647
648     foreach (var body in data)
649     {
650         if (body == null)
651         {

```

```

652         continue;
653     }
654
655     if (body.IsTracked)
656     {
657         if (!_Bodies.ContainsKey(body.TrackingId))
658         {
659             _Bodies[body.TrackingId] = CreateBodyObject
660                 (body.TrackingId);
661
662             RefreshBodyObject(body, _Bodies[body.TrackingId]);
663
664             Quaternion SpineBase = body.JointOrientations
665                 [JointType.SpineBase].Orientation.ChangeQuat(newrotation);
666             Quaternion SpineMid = body.JointOrientations
667                 [JointType.SpineMid].Orientation.ChangeQuat(newrotation);
668             Quaternion SpineShoulder = body.JointOrientations
669                 [JointType.SpineShoulder].Orientation.ChangeQuat
670                 (newrotation);
671             Quaternion ShoulderLeft = body.JointOrientations
672                 [JointType.ShoulderLeft].Orientation.ChangeQuat
673                 (newrotation);
674             Quaternion ShoulderRight = body.JointOrientations
675                 [JointType.ShoulderRight].Orientation.ChangeQuat
676                 (newrotation);
677             Quaternion ElbowLeft = body.JointOrientations
678                 [JointType.ElbowLeft].Orientation.ChangeQuat(newrotation);
679             Quaternion WristLeft = body.JointOrientations
680                 [JointType.WristLeft].Orientation.ChangeQuat(newrotation);
681             Quaternion HandLeft = body.JointOrientations
682                 [JointType.HandLeft].Orientation.ChangeQuat(newrotation);
683             Quaternion ElbowRight = body.JointOrientations
684                 [JointType.ElbowRight].Orientation.ChangeQuat(newrotation);
685             Quaternion WristRight = body.JointOrientations
686                 [JointType.WristRight].Orientation.ChangeQuat(newrotation);
687             Quaternion HandRight = body.JointOrientations
688                 [JointType.HandRight].Orientation.ChangeQuat(newrotation);
689             Quaternion KneeLeft = body.JointOrientations
690                 [JointType.KneeLeft].Orientation.ChangeQuat(newrotation);
691             Quaternion AnkleLeft = body.JointOrientations
692                 [JointType.AnkleLeft].Orientation.ChangeQuat(newrotation);
693             Quaternion KneeRight = body.JointOrientations
694                 [JointType.KneeRight].Orientation.ChangeQuat(newrotation);
695             Quaternion AnkleRight = body.JointOrientations
696                 [JointType.AnkleRight].Orientation.ChangeQuat(newrotation);
697             //Quaternion LegRight = body.JointOrientations
698                 [JointType.RightLeg].Orientation.ChangeQuat
699                 (newrotation); //
700             //Quaternion LegLeft = body.JointOrientations
701                 [JointType.LeftLeg].Orientation.ChangeQuat(newrotation); //
702             Quaternion FootRight = body.JointOrientations
703                 [JointType.FootRight].Orientation.ChangeQuat
704                 (newrotation); //
705             Quaternion FootLeft = body.JointOrientations
706                 [JointType.FootLeft].Orientation.ChangeQuat

```

```

        (newrotation); //
683
684     FootRight.y = 1;
685     FootRight.w = 0;
686     FootLeft.y = 1;
687     FootLeft.w = 0;
688     Spine1.transform.rotation = SpineMid * Quaternion.AngleAxis
        (-90, new Vector3(0, 0, 1));
689     RightArm.transform.rotation = ElbowRight *
        Quaternion.AngleAxis(-90, new Vector3(0, 0, 1));
690     RightForeArm.transform.rotation = WristRight *
        Quaternion.AngleAxis(-90, new Vector3(0, 0, 1));
691     RightHand.transform.rotation = HandRight *
        Quaternion.AngleAxis(-90, new Vector3(0, 0, 1));
692     LeftArm.transform.rotation = ElbowLeft * Quaternion.AngleAxis
        (180, new Vector3(0, 1, 0)) *
        Quaternion.AngleAxis(90, new Vector3(0, 0, 1));
693     LeftForeArm.transform.rotation = WristLeft *
        Quaternion.AngleAxis(180, new Vector3(0, 1, 0)) *
        Quaternion.AngleAxis(90, new Vector3(0, 0, 1));
694     RightFoot.transform.rotation = FootRight *
        Quaternion.AngleAxis(0, new Vector3(0, 0, 1)) *
        Quaternion.AngleAxis(-90, new Vector3(0, 1, 0)); // ojo -90
697
698     LeftFoot.transform.rotation = FootLeft * Quaternion.AngleAxis
        (180, new Vector3(1, 0, 0)) * Quaternion.AngleAxis(-90, new
        Vector3(0, 1, 0)); //
699     LeftHand.transform.rotation = HandLeft * Quaternion.AngleAxis
        (180, new Vector3(0, 1, 0)) * Quaternion.AngleAxis(90, new
        Vector3(0, 0, 1));
700     RightUpLeg.transform.rotation = KneeRight *
        Quaternion.AngleAxis(90, new Vector3(0, 1, 0)) *
        Quaternion.AngleAxis(-90, new Vector3(0, 0, 1));
701     RightLeg.transform.rotation = AnkleRight *
        Quaternion.AngleAxis(90, new Vector3(0, 1, 0)) *
        Quaternion.AngleAxis(-90, new Vector3(0, 0, 1));
702     LeftUpLeg.transform.rotation = KneeLeft *
        Quaternion.AngleAxis(90, new Vector3(0, 1, 0)) *
        Quaternion.AngleAxis(90, new Vector3(0, 0, 1));
703     LeftLeg.transform.rotation = AnkleLeft * Quaternion.AngleAxis
        (90, new Vector3(0, 1, 0)) * Quaternion.AngleAxis(90, new
        Vector3(0, 0, 1));
704
705     var moveposition = body.Joints[JointType.SpineMid].Position;
706     Master.transform.position = new Vector3(moveposition.X,
        moveposition.Y, -moveposition.Z);
707
708
709
710     //////////////////////////////////superior
711
712     // :::::::::::::::::::::: HORIZONTAL ARMS
        EXERCISE :::::::::::::::::::::: //
713
714     if (Input.GetKeyDown(KeyCode.Q))
715     {

```



```

716         Debug.Log("Horizontal Exercise selected");
717         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
718             " ::::: HORIZONTAL EXERCISE ::::: ");
719         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
720             Environment.NewLine);
721     }
722     // :::::::::::::: TOUCH SHOULDERS
723     EXERCISE :::::::::::::: //
724     if (Input.GetKeyDown(KeyCode.W))
725     {
726         Debug.Log("Touch Shoulders Exercise selected");
727         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
728             " ::::: TOUCH SHOULDERS EXERCISE ::::: ");
729         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
730             Environment.NewLine);
731     }
732     // :::::::::::::: ARMS UP
733     EXERCISE :::::::::::::: //
734     if (Input.GetKeyDown(KeyCode.E))
735     {
736         Debug.Log("Arms Up Exercise Selected");
737         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
738             " ::::: ARMS UP EXERCISE ::::: ");
739         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
740             Environment.NewLine);
741     }
742     // :::::::::::::: FLEX :::::::::::::: //
743     if (flex == true)
744     {
745         //variables
746         float x1 = body.Joints
747             [JointType.ShoulderRight].Position.X;
748         float y1 = body.Joints
749             [JointType.ShoulderRight].Position.Y;
750         float z1 = body.Joints
751             [JointType.ShoulderRight].Position.Z;
752         float x2 = body.Joints
753             [JointType.ShoulderLeft].Position.X;
754         float y2 = body.Joints
755             [JointType.ShoulderLeft].Position.Y;
756         float z2 = body.Joints
757             [JointType.ShoulderLeft].Position.Z;
758         float x3 = body.Joints[JointType.SpineMid].Position.X;
759         float y3 = body.Joints[JointType.SpineMid].Position.Y;
760         float z3 = body.Joints[JointType.SpineMid].Position.Z;
761         float u1 = body.Joints[JointType.SpineBase].Position.X;
762         float u2 = body.Joints[JointType.SpineBase].Position.Y;
763         float u3 = body.Joints[JointType.SpineBase].Position.Z;

```

```

758         float c1 = body.Joints[JointType.ElbowRight].Position.X;
759         float c2 = body.Joints[JointType.ElbowRight].Position.Y;
760         float c3 = body.Joints[JointType.ElbowRight].Position.Z;
761         float c1l = body.Joints[JointType.ElbowLeft].Position.X;
762         float c2l = body.Joints[JointType.ElbowLeft].Position.Y;
763         float c3l = body.Joints[JointType.ElbowLeft].Position.Z;
764
765         //planes
766         Vector3 vector1 = new Vector3(x2 - x1, y2 - y1, z2 - z1);
767         Vector3 vector2 = new Vector3(x3 - x1, y3 - y1, z3 - z1);
768         Vector3 cp;
769         cp = Vector3.Cross(vector1, vector2);
770         Vector3 normalvect = new Vector3(cp.x, cp.y, cp.z);
771         Vector3 auxvect2 = new Vector3(x3 - u1, y3 - u2, z3 -
772         u3);
773         Vector3 newcp;
774         newcp = Vector3.Cross(normalvect, auxvect2);
775         float daux = newcp.x * (-x3) + newcp.y * (-y3) + newcp.z
776         * (-z3);
777
778         //right side projection and result
779         float lambdaaux = (((-newcp.x) * c1 + (-newcp.y) * c2 +
780         (-newcp.z) * c3) - daux) / ((newcp.x) * (newcp.x) +
781         (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
782         Vector3 elbowrproject = new Vector3(c1 + (lambdaaux *
783         newcp.x), c2 + (lambdaaux * newcp.y), c3 + (lambdaaux *
784         newcp.z));
785         float shoulderXr = body.Joints
786         [JointType.ShoulderRight].Position.X;
787         float shoulderYr = body.Joints
788         [JointType.ShoulderRight].Position.Y;
789         float shoulderZr = body.Joints
790         [JointType.ShoulderRight].Position.Z;
791         float lambdaaux1 = (((-newcp.x) * shoulderXr + (-newcp.y)
792         * shoulderYr + (-newcp.z) * shoulderZr) - daux) /
793         ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z)
794         * (newcp.z));
795         Vector3 shoulderproject = new Vector3(shoulderXr +
796         (lambdaaux1 * newcp.x), shoulderYr + (lambdaaux1 *
797         newcp.y), shoulderZr + (lambdaaux1 * newcp.z));
798         Vector3 vectorelbowshould = new Vector3(shoulderproject.x
799         - elbowrproject.x, shoulderproject.y - elbowrproject.y,
800         shoulderproject.z - elbowrproject.z);
801         float angleFlex = Vector3.Angle(auxvect2,
802         vectorelbowshould);
803
804         //Debug.Log("MAX ELEMENT OF THE ARRAY! " +Mathf.Max
805         (RightFlex) + " " + RightFlex.Max());
806
807         //left side projection and result
808         float lambdaaux2 = (((-newcp.x) * c1l + (-newcp.y) * c2l
809         + (-newcp.z) * c3l) - daux) / ((newcp.x) * (newcp.x) +

```

```

795         (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
Vector3 elbowrprojectl = new Vector3(c1l + (lambdaaux2 *
796         newcp.x), c2l + (lambdaaux2 * newcp.y), c3l + (lambdaaux2 *
newcp.z));
797         float shoulderXl = body.Joints
[JointType.ShoulderLeft].Position.X;
798         float shoulderYl = body.Joints
[JointType.ShoulderRight].Position.Y;
799         float shoulderZl = body.Joints
[JointType.ShoulderRight].Position.Z;
800         float lambdaaux3 = (((-newcp.x) * shoulderXl + (-newcp.y) *
shoulderYl + (-newcp.z) * shoulderZl) - daux) /
801         (((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) *
(newcp.z)));
Vector3 shoulderprojectl = new Vector3(shoulderXl +
802         (lambdaaux3 * newcp.x), shoulderYl + (lambdaaux3 *
newcp.y), shoulderZl + (lambdaaux3 * newcp.z));
803         Vector3 vectorelbowshouldl = new Vector3
(shoulderprojectl.x - elbowrprojectl.x, shoulderprojectl.y -
804         elbowrprojectl.y, shoulderprojectl.z - elbowrprojectl.z);
float angleFlexl = Vector3.Angle(auxvect2,
805         vectorelbowshouldl);
float anglecorrectl = 180 - angleFlexl;
806         //Debug.Log("Angle Left Flex:" + angleFlexl);
StartCoroutine(Temporizador(angleFlex, angleFlexl));
807
808         //File results
809         // File.AppendAllText(@"C:\Users\Aleix\Desktop
\Results.txt", " Angle Right Flex: " + angleFlex + "
810         Angle Left Flex: " + angleFlexl);
// File.AppendAllText(@"C:\Users\Aleix\Desktop
\Results.txt", Environment.NewLine);
811     }
812
813     // :::::::::::::::::::: ABD :::::::::::::::::::: //
814
815     if (ABDUP==true)
816     {
817
818         //variables
819         float x1 = body.Joints
[JointType.ShoulderRight].Position.X;
820         float y1 = body.Joints
[JointType.ShoulderRight].Position.Y;
821         float z1 = body.Joints
[JointType.ShoulderRight].Position.Z;
822         Vector3 shoulderright = new Vector3(x1, y1, z1);
823         float x2 = body.Joints
[JointType.ShoulderLeft].Position.X;
824         float y2 = body.Joints
[JointType.ShoulderLeft].Position.Y;
825         float z2 = body.Joints
[JointType.ShoulderLeft].Position.Z;
826         Vector3 shoulderleft = new Vector3(x2, y2, z2);

```

```

827         float x3 = body.Joints[JointType.SpineMid].Position.X;
828         float y3 = body.Joints[JointType.SpineMid].Position.Y;
829         float z3 = body.Joints[JointType.SpineMid].Position.Z;
830         Vector3 spinemidv = new Vector3(x3, y3, z3);
831         float c1 = body.Joints[JointType.ElbowRight].Position.X;
832         float c2 = body.Joints[JointType.ElbowRight].Position.Y;
833         float c3 = body.Joints[JointType.ElbowRight].Position.Z;
834         Vector3 elbowrightv = new Vector3(c1, c2, c3);
835
836         //plane
837         Vector3 vector1 = new Vector3(x2 - x1, y2 - y1, z2 - z1);
838         Vector3 vector2 = new Vector3(x3 - x1, y3 - y1, z3 - z1);
839         Vector3 cp;
840         cp = Vector3.Cross(vector1, vector2);
841         float d = cp.x * (-x1) + cp.y * (-y1) + cp.z * (-z1);
842
843         //right side projection and result
844         float lambda = (((-cp.x) * c1 + (-cp.y) * c2 + (-cp.z) *
845         c3) - d) / ((cp.x) * (cp.x) + (cp.y) * (cp.y) + (cp.z) *
846         (cp.z));
847         float valor1 = (-cp.x) * c1 + (-cp.y) * c2 + (-cp.z) *
848         c3;
849         float valor2 = (cp.x * cp.x + cp.y * cp.y + cp.z * cp.z);
850         Vector3 cprojected = new Vector3(c1 + (lambda * cp.x), c2
851         + (lambda * cp.y), c3 + (lambda * cp.z));
852         Vector3 vectorfinalelbow = new Vector3(cprojected.x - x1,
853         cprojected.y - y1, cprojected.z - z1);
854         float u1 = body.Joints[JointType.SpineBase].Position.X;
855         float u2 = body.Joints[JointType.SpineBase].Position.Y;
856         float u3 = body.Joints[JointType.SpineBase].Position.Z;
857         float lambda3 = (((-cp.x * u1 - cp.y * u2 - cp.z * u3) -
858         d) / (cp.x * cp.x + cp.y * cp.y + cp.z * cp.z));
859         Vector3 spinebaseproj = new Vector3(u1 + lambda3 * cp.x,
860         u2 + lambda3 * cp.y, u3 + lambda3 * cp.z);
861         Vector3 vectorhorizproj = new Vector3(x3 -
862         spinebaseproj.x, y3 - spinebaseproj.y, z3 -
863         spinebaseproj.z);
864         float angleABDfinal = Vector3.Angle(vectorfinalelbow,
865         vectorhorizproj);
866         float correct = 180 - angleABDfinal;
867         //Debug.Log("Angle ABD right = " + correct);
868
869         //Left side projection and result
870         float c1l = body.Joints[JointType.ElbowLeft].Position.X;
871         float c2l = body.Joints[JointType.ElbowLeft].Position.Y;
872         float c3l = body.Joints[JointType.ElbowLeft].Position.Z;
873         float lambdal = (((-cp.x) * c1l + (-cp.y) * c2l + (-cp.z) *
874         c3l) - d) / ((cp.x) * (cp.x) + (cp.y) * (cp.y) + (cp.z) *
875         (cp.z));
876         float valor1l = (-cp.x) * c1l + (-cp.y) * c2l + (-cp.z) *
877         c3l;
878         float valor2l = (cp.x * cp.x + cp.y * cp.y + cp.z *
879         cp.z);
880         Vector3 clprojected = new Vector3(c1l + (lambdal * cp.x),
881         c2l + (lambdal * cp.y), c3l + (lambdal * cp.z));

```

```

868         Vector3 clandelbowleft = new Vector3(clprojected.x - x2, ↗
clprojected.y - y2, clprojected.z - z2);
869         float angleABDlfinal = Vector3.Angle(clandelbowleft, ↗
vectorhorizproj);
870         float correctl = 180 - angleABDlfinal;
871         // Debug.Log("Angle Abd Left: " + correctl);
872
873         //File results
874
875         //File.AppendAllText(@"C:\Users\Aleix\Desktop ↗
\Results.txt", " Angle Right ABD: " + correct + " Angle ↗
Left ABD: " + correctl);
876         //File.AppendAllText(@"C:\Users\Aleix\Desktop ↗
\Results.txt", Environment.NewLine);
877
878         StartCoroutine(TemporizadorABDUP(correct, correctl));
879         //flex
880         //new plane
881         Vector3 normalvect = new Vector3(cp.x, cp.y, cp.z);
882         Vector3 auxvect2 = new Vector3(x3 - u1, y3 - u2, z3 - ↗
u3);
883         Vector3 newcp;
884         newcp = Vector3.Cross(normalvect, auxvect2);
885         float daux = newcp.x * (-x3) + newcp.y * (-y3) + newcp.z ↗
* (-z3);
886         //elbow right point and projection
887         float lambdaaux = (((-newcp.x) * c1 + (-newcp.y) * c2 + ↗
(-newcp.z) * c3) - daux) / ((newcp.x) * (newcp.x) + ↗
(newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
888         Vector3 elbowrproject = new Vector3(c1 + (lambdaaux * ↗
newcp.x), c2 + (lambdaaux * newcp.y), c3 + (lambdaaux * ↗
newcp.z));
889         //shoulder right point and projection
890         float shoulderXr = body.Joints ↗
[JointType.ShoulderRight].Position.X;
891         float shoulderYr = body.Joints ↗
[JointType.ShoulderRight].Position.Y;
892         float shoulderZr = body.Joints ↗
[JointType.ShoulderRight].Position.Z;
893         float lambdaaux1 = (((-newcp.x) * shoulderXr + (-newcp.y) ↗
* shoulderYr + (-newcp.z) * shoulderZr) - daux) / ↗
((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) ↗
* (newcp.z));
894         Vector3 shoulderproject = new Vector3(shoulderXr + ↗
(lambdaaux1 * newcp.x), shoulderYr + (lambdaaux1 * ↗
newcp.y), shoulderZr + (lambdaaux1 * newcp.z));
895
896         Vector3 vectorelbowshould = new Vector3(shoulderproject.x ↗
- elbowrproject.x, shoulderproject.y - elbowrproject.y, ↗
shoulderproject.z - elbowrproject.z);
897
898         float angleFlex = Vector3.Angle(auxvect2, ↗
vectorelbowshould);
899         //Debug.Log("Angle flex right:" + angleFlex);
900
901         //left

```

```

902         float lambdaaux2 = (((-newcp.x) * c1l + (-newcp.y) * c2l +
+ (-newcp.z) * c3l) - daux) / ((newcp.x) * (newcp.x) +
+ (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
903         Vector3 elbowrprojectl = new Vector3(c1l + (lambdaaux2 *
+ newcp.x), c2l + (lambdaaux2 * newcp.y), c3l + (lambdaaux2
+ * newcp.z));
904         //shoulder left point and projection
905         float shoulderXl = body.Joints
+ [JointType.ShoulderLeft].Position.X;
906         float shoulderYl = body.Joints
+ [JointType.ShoulderRight].Position.Y;
907         float shoulderZl = body.Joints
+ [JointType.ShoulderRight].Position.Z;
908         float lambdaaux3 = (((-newcp.x) * shoulderXl + (-newcp.y)
+ * shoulderYl + (-newcp.z) * shoulderZl) - daux) /
+ ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z)
+ * (newcp.z));
909         Vector3 shoulderprojectl = new Vector3(shoulderXl +
+ (lambdaaux3 * newcp.x), shoulderYl + (lambdaaux3 *
+ newcp.y), shoulderZl + (lambdaaux3 * newcp.z));
910         Vector3 vectorelbowshouldl = new Vector3
+ (shoulderprojectl.x - elbowrprojectl.x, shoulderprojectl.y
+ - elbowrprojectl.y, shoulderprojectl.z -
+ elbowrprojectl.z);
911         float angleFlexl = Vector3.Angle(auxvect2,
+ vectorelbowshouldl);
912         // Debug.Log("Angle flex no correction left:" +
+ angleFlexl);
913
914
915     }
916
917     if (ABDUP == true)
918     {
919
920         //variables
921         float x1 = body.Joints
+ [JointType.ShoulderRight].Position.X;
922         float y1 = body.Joints
+ [JointType.ShoulderRight].Position.Y;
923         float z1 = body.Joints
+ [JointType.ShoulderRight].Position.Z;
924         Vector3 shoulderright = new Vector3(x1, y1, z1);
925         float x2 = body.Joints
+ [JointType.ShoulderLeft].Position.X;
926         float y2 = body.Joints
+ [JointType.ShoulderLeft].Position.Y;
927         float z2 = body.Joints
+ [JointType.ShoulderLeft].Position.Z;
928         Vector3 shoulderleft = new Vector3(x2, y2, z2);
929         float x3 = body.Joints[JointType.SpineMid].Position.X;
930         float y3 = body.Joints[JointType.SpineMid].Position.Y;
931         float z3 = body.Joints[JointType.SpineMid].Position.Z;
932         Vector3 spinemidv = new Vector3(x3, y3, z3);
933         float c1 = body.Joints[JointType.ElbowRight].Position.X;
934         float c2 = body.Joints[JointType.ElbowRight].Position.Y;

```

```

935         float c3 = body.Joints[JointType.ElbowRight].Position.Z;
936         Vector3 elbowrightv = new Vector3(c1, c2, c3);
937
938         //plane
939         Vector3 vector1 = new Vector3(x2 - x1, y2 - y1, z2 - z1);
940         Vector3 vector2 = new Vector3(x3 - x1, y3 - y1, z3 - z1);
941         Vector3 cp;
942         cp = Vector3.Cross(vector1, vector2);
943         float d = cp.x * (-x1) + cp.y * (-y1) + cp.z * (-z1);
944
945         //right side projection and result
946         float lambda = (((-cp.x) * c1 + (-cp.y) * c2 + (-cp.z) *
947         c3) - d) / ((cp.x) * (cp.x) + (cp.y) * (cp.y) + (cp.z) *
948         (cp.z));
949         float valor1 = (-cp.x) * c1 + (-cp.y) * c2 + (-cp.z) *
950         c3;
951         float valor2 = (cp.x * cp.x + cp.y * cp.y + cp.z * cp.z);
952         Vector3 cprojected = new Vector3(c1 + (lambda * cp.x), c2
953         + (lambda * cp.y), c3 + (lambda * cp.z));
954         Vector3 vectorfinalelbow = new Vector3(cprojected.x - x1,
955         cprojected.y - y1, cprojected.z - z1);
956         float u1 = body.Joints[JointType.SpineBase].Position.X;
957         float u2 = body.Joints[JointType.SpineBase].Position.Y;
958         float u3 = body.Joints[JointType.SpineBase].Position.Z;
959         float lambda3 = (((-cp.x * u1 - cp.y * u2 - cp.z * u3) -
960         d) / (cp.x * cp.x + cp.y * cp.y + cp.z * cp.z));
961         Vector3 spinebaseproj = new Vector3(u1 + lambda3 * cp.x,
962         u2 + lambda3 * cp.y, u3 + lambda3 * cp.z);
963         Vector3 vectorhorizproj = new Vector3(x3 -
964         spinebaseproj.x, y3 - spinebaseproj.y, z3 -
965         spinebaseproj.z);
966         float angleABDfinal = Vector3.Angle(vectorfinalelbow,
967         vectorhorizproj);
968         float correct = 180 - angleABDfinal;
969         //Debug.Log("Angle ABD right = " + correct);
970
971         //Left side projection and result
972         float c1l = body.Joints[JointType.ElbowLeft].Position.X;
973         float c2l = body.Joints[JointType.ElbowLeft].Position.Y;
974         float c3l = body.Joints[JointType.ElbowLeft].Position.Z;
975         float lambdal = (((-cp.x) * c1l + (-cp.y) * c2l + (-cp.z) *
976         c3l) - d) / ((cp.x) * (cp.x) + (cp.y) * (cp.y) + (cp.z) *
977         (cp.z));
978         float valor1l = (-cp.x) * c1l + (-cp.y) * c2l + (-cp.z) *
979         c3l;
980         float valor2l = (cp.x * cp.x + cp.y * cp.y + cp.z *
981         cp.z);
982         Vector3 clprojected = new Vector3(c1l + (lambdal * cp.x),
983         c2l + (lambdal * cp.y), c3l + (lambdal * cp.z));
984         Vector3 clandelbowleft = new Vector3(clprojected.x - x2,
985         clprojected.y - y2, clprojected.z - z2);
986         float angleABDlfinal = Vector3.Angle(clandelbowleft,
987         vectorhorizproj);
988         float correctl = 180 - angleABDlfinal;
989         // Debug.Log("Angle Abd Left: " + correctl);

```



```

974
975         //File results
976
977         //File.AppendAllText(@"C:\Users\Aleix\Desktop
\Results.txt", " Angle Right ABD: " + correct + " Angle
Left ABD: " + correctl);
978         //File.AppendAllText(@"C:\Users\Aleix\Desktop
\Results.txt", Environment.NewLine);
979
980         StartCoroutine(TemporizadorABDUP(correct, correctl));
981         //flex
982         //new plane
983         Vector3 normalvect = new Vector3(cp.x, cp.y, cp.z);
984         Vector3 auxvect2 = new Vector3(x3 - u1, y3 - u2, z3 -
u3);
985         Vector3 newcp;
986         newcp = Vector3.Cross(normalvect, auxvect2);
987         float daux = newcp.x * (-x3) + newcp.y * (-y3) + newcp.z
* (-z3);
988         //elbow right point and projection
989         float lambdaaux = (((-newcp.x) * c1 + (-newcp.y) * c2 +
(-newcp.z) * c3) - daux) / ((newcp.x) * (newcp.x) +
(newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
990         Vector3 elbowrproject = new Vector3(c1 + (lambdaaux *
newcp.x), c2 + (lambdaaux * newcp.y), c3 + (lambdaaux *
newcp.z));
991         //shoulder right point and projection
992         float shoulderXr = body.Joints
[JointType.ShoulderRight].Position.X;
993         float shoulderYr = body.Joints
[JointType.ShoulderRight].Position.Y;
994         float shoulderZr = body.Joints
[JointType.ShoulderRight].Position.Z;
995         float lambdaaux1 = (((-newcp.x) * shoulderXr + (-newcp.y)
* shoulderYr + (-newcp.z) * shoulderZr) - daux) /
((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z)
* (newcp.z));
996         Vector3 shoulderproject = new Vector3(shoulderXr +
(lambdaaux1 * newcp.x), shoulderYr + (lambdaaux1 *
newcp.y), shoulderZr + (lambdaaux1 * newcp.z));
997
998         Vector3 vectorelbowshould = new Vector3(shoulderproject.x
- elbowrproject.x, shoulderproject.y - elbowrproject.y,
shoulderproject.z - elbowrproject.z);
999
1000         float angleFlex = Vector3.Angle(auxvect2,
vectorelbowshould);
1001         //Debug.Log("Angle flex right:" + angleFlex);
1002
1003         //left
1004         float lambdaaux2 = (((-newcp.x) * c1l + (-newcp.y) * c2l
+ (-newcp.z) * c3l) - daux) / ((newcp.x) * (newcp.x) +
(newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
1005         Vector3 elbowrprojectl = new Vector3(c1l + (lambdaaux2 *
newcp.x), c2l + (lambdaaux2 * newcp.y), c3l + (lambdaaux2
* newcp.z));

```



```

1006         //shoulder left point and projection
1007         float shoulderXl = body.Joints
[JointType.ShoulderLeft].Position.X;
1008         float shoulderYl = body.Joints
[JointType.ShoulderRight].Position.Y;
1009         float shoulderZl = body.Joints
[JointType.ShoulderRight].Position.Z;
1010         float lambdaaux3 = (((-newcp.x) * shoulderXl + (-newcp.y)
* shoulderYl + (-newcp.z) * shoulderZl) - daux) /
((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z)
* (newcp.z));
1011         Vector3 shoulderprojectl = new Vector3(shoulderXl +
(lambdaaux3 * newcp.x), shoulderYl + (lambdaaux3 *
newcp.y), shoulderZl + (lambdaaux3 * newcp.z));
1012         Vector3 vectorelbowshouldl = new Vector3
(shoulderprojectl.x - elbowrprojectl.x, shoulderprojectl.y
- elbowrprojectl.y, shoulderprojectl.z -
elbowrprojectl.z);
1013         float angleFlexl = Vector3.Angle(auxvect2,
vectorelbowshouldl);
1014         // Debug.Log("Angle flex no correction left:" +
angleFlexl);
1015
1016
1017     }
1018
1019     // :::::::::::::::::::: ELBOW
EXTENSION :::::::::::::::::::: //
1020
1021
1022     if (Elbow==true)
1023     {
1024         //variables
1025         float shoulderXr = body.Joints
[JointType.ShoulderRight].Position.X;
1026         float shoulderYr = body.Joints
[JointType.ShoulderRight].Position.Y;
1027         float shoulderZr = body.Joints
[JointType.ShoulderRight].Position.Z;
1028         float relbowX = body.Joints
[JointType.ElbowRight].Position.X;
1029         float relbowY = body.Joints
[JointType.ElbowRight].Position.Y;
1030         float relbowZ = body.Joints
[JointType.ElbowRight].Position.Z;
1031         float wristXr = body.Joints
[JointType.WristRight].Position.X;
1032         float wristYr = body.Joints
[JointType.WristRight].Position.Y;
1033         float wristZr = body.Joints
[JointType.WristRight].Position.Z;
1034         float shoulderXl = body.Joints
[JointType.ShoulderLeft].Position.X;
1035         float shoulderYl = body.Joints
[JointType.ShoulderLeft].Position.Y;
1036         float shoulderZl = body.Joints

```

```

1037     [JointType.ShoulderLeft].Position.Z;
1038     float lelbowX = body.Joints
1039     [JointType.ElbowLeft].Position.X;
1040     float lelbowY = body.Joints
1041     [JointType.ElbowLeft].Position.Y;
1042     float lelbowZ = body.Joints
1043     [JointType.ElbowLeft].Position.Z;
1044     float wristXl = body.Joints
1045     [JointType.WristLeft].Position.X;
1046     float wristYl = body.Joints
1047     [JointType.WristLeft].Position.Y;
1048     float wristZl = body.Joints
1049     [JointType.WristLeft].Position.Z;
1050
1051     //right side results
1052     Vector3 vecSandEr = new Vector3(shoulderXr - relbowX,
1053     shoulderYr - relbowY, shoulderZr - relbowZ);
1054     Vector3 vecWandEr = new Vector3(wristXr - relbowX,
1055     wristYr - relbowY, wristZr - relbowZ);
1056     float angleElbowr = Vector3.Angle(vecSandEr, vecWandEr);
1057     Debug.Log("New angle right elbow: " + angleElbowr);
1058
1059     //left side results
1060     Vector3 vecSandEl = new Vector3(shoulderXl - lelbowX,
1061     shoulderYl - lelbowY, shoulderZl - lelbowZ);
1062     Vector3 vecWandEl = new Vector3(wristXl - lelbowX,
1063     wristYl - lelbowY, wristZl - lelbowZ);
1064     float angleElbowl = Vector3.Angle(vecSandEl, vecWandEl);
1065     Debug.Log("New angle left elbow: " + angleElbowl);
1066
1067     //File results
1068     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
1069     " Right Elbow Extension: " + angleElbowr +
1070     " Left Elbow Extension: " + angleElbowl);
1071     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
1072     Environment.NewLine);
1073
1074 }
1075
1076 if (HABDUP == true)
1077 {
1078     float x1 = body.Joints
1079     [JointType.ShoulderRight].Position.X;
1080     float y1 = body.Joints
1081     [JointType.ShoulderRight].Position.Y;
1082     float z1 = body.Joints
1083     [JointType.ShoulderRight].Position.Z;
1084
1085     float x2 = body.Joints
1086     [JointType.ShoulderLeft].Position.X;
1087     float y2 = body.Joints
1088     [JointType.ShoulderLeft].Position.Y;
1089     float z2 = body.Joints
1090     [JointType.ShoulderLeft].Position.Z;
1091
1092     float a1 = body.Joints[JointType.WristRight].Position.X;

```

```

1074         float b1 = body.Joints[JointType.WristRight].Position.Y;
1075         float c1 = body.Joints[JointType.WristRight].Position.Z;
1076
1077         float a2 = body.Joints[JointType.WristLeft].Position.X;
1078         float b2 = body.Joints[JointType.WristLeft].Position.Y;
1079         float c2 = body.Joints[JointType.WristLeft].Position.Z;
1080
1081         float a3 = body.Joints[JointType.SpineMid].Position.X;
1082         float b3 = body.Joints[JointType.SpineMid].Position.Y;
1083         float c3 = body.Joints[JointType.SpineMid].Position.Z;
1084
1085         float a4 = body.Joints[JointType.ElbowRight].Position.X;
1086         float b4 = body.Joints[JointType.ElbowRight].Position.Y;
1087         float c4 = body.Joints[JointType.ElbowRight].Position.Z;
1088
1089         float a5 = body.Joints[JointType.ElbowLeft].Position.X;
1090         float b5 = body.Joints[JointType.ElbowLeft].Position.Y;
1091         float c5 = body.Joints[JointType.ElbowLeft].Position.Z;
1092
1093         float a6 = body.Joints[JointType.SpineBase].Position.X;
1094         float b6 = body.Joints[JointType.SpineBase].Position.Y;
1095         float c6 = body.Joints[JointType.SpineBase].Position.Z;
1096
1097         float a7 = body.Joints[JointType.Neck].Position.X;
1098         float b7 = body.Joints[JointType.Neck].Position.Y;
1099         float c7 = body.Joints[JointType.Neck].Position.Z;
1100
1101         float a8 = body.Joints[JointType.Head].Position.X;
1102         float b8 = body.Joints[JointType.Head].Position.Y;
1103         float c8 = body.Joints[JointType.Head].Position.Z;
1104
1105
1106         Vector3 shoulderright = new Vector3(x1, y1, z1);
1107         Vector3 shoulderleft = new Vector3(x2, y2, z2);
1108         Vector3 wristright = new Vector3(a1, b1, c1);
1109         Vector3 wristleft = new Vector3(a2, b2, c2);
1110         Vector3 spinemidv = new Vector3(a3, b3, c3);
1111         Vector3 Elbowright = new Vector3(a4, b4, c4);
1112         Vector3 Elbowleft = new Vector3(a5, b5, c5);
1113         Vector3 SpineB = new Vector3(a6, b6, c6);
1114         Vector3 neck = new Vector3(a7, b7, c7);
1115         Vector3 head = new Vector3(a8, b8, c8);
1116
1117         Vector3 shoulders = new Vector3(x2 - x1, y2 - y1, z2 -
1118         z1);
1119         Vector3 rightarm = new Vector3(a4 - x1, b4 - y1, c4 -
1120         z1);
1121         Vector3 leftarm = new Vector3(a5 - x2, b5 - y2, c5 - z2);
1122
1123         Plane Chest = new Plane();
1124         Chest.Set3Points(shoulderright, shoulderleft, spinemidv);
1125         Vector3 ChestVector = new Vector3();
1126         ChestVector = -Chest.normal;
1127         float HABDUPleft = Vector3.Angle(ChestVector, leftarm);
1128
1129         float HABDUPright = Vector3.Angle(ChestVector, rightarm);

```

```

1127         float FLEXleft = Vector3.Angle(-shoulders, leftarm);
1128         float FLEXright = Vector3.Angle(shoulders, rightarm);
1129
1130         if (Mathf.Abs(HABDUPright) < 30 && rep == true)
1131         {
1132             if (repeticiones0 == true)
1133             {
1134                 conta = conta + 1;
1135             }
1136             rep = false;
1137         }
1138         if (Mathf.Abs(HABDUPright) > 30)
1139         {
1140             rep = true;
1141             repeticiones0 = true;
1142         }
1143         var exp = repeticionesHABDUP.GetComponent<Text>();
1144         exp.text = (""+conta);
1145         repHABDUP = exp.text;
1146
1147         StartCoroutine(TemporizadorHABDUP(HABDUPleft,
1148         HABDUPright, FLEXleft, FLEXright,
1149         shoulderright,shoulderleft,wristright,wristright,spinemidv
1150         , Elbowright,Elbowleft, SpineB,neck,head, HABDUPtime));
1151
1152     }
1153
1154     if (bomba.PlayBomb == 1)
1155     {
1156         float x1 = body.Joints
1157         [JointType.ShoulderRight].Position.X;
1158         float y1 = body.Joints
1159         [JointType.ShoulderRight].Position.Y;
1160         float z1 = body.Joints
1161         [JointType.ShoulderRight].Position.Z;
1162
1163         float x2 = body.Joints
1164         [JointType.ShoulderLeft].Position.X;
1165         float y2 = body.Joints
1166         [JointType.ShoulderLeft].Position.Y;
1167         float z2 = body.Joints
1168         [JointType.ShoulderLeft].Position.Z;
1169
1170         float a1 = body.Joints[JointType.WristRight].Position.X;
1171         float b1 = body.Joints[JointType.WristRight].Position.Y;
1172         float c1 = body.Joints[JointType.WristRight].Position.Z;
1173
1174         float a2 = body.Joints[JointType.WristLeft].Position.X;
1175         float b2 = body.Joints[JointType.WristLeft].Position.Y;
1176         float c2 = body.Joints[JointType.WristLeft].Position.Z;
1177
1178         float a3 = body.Joints[JointType.SpineMid].Position.X;
1179         float b3 = body.Joints[JointType.SpineMid].Position.Y;
1180         float c3 = body.Joints[JointType.SpineMid].Position.Z;
1181
1182         float a4 = body.Joints[JointType.ElbowRight].Position.X;

```

```

1174         float b4 = body.Joints[JointType.ElbowRight].Position.Y;
1175         float c4 = body.Joints[JointType.ElbowRight].Position.Z;
1176
1177         float a5 = body.Joints[JointType.ElbowLeft].Position.X;
1178         float b5 = body.Joints[JointType.ElbowLeft].Position.Y;
1179         float c5 = body.Joints[JointType.ElbowLeft].Position.Z;
1180
1181         float a6 = body.Joints[JointType.SpineBase].Position.X;
1182         float b6 = body.Joints[JointType.SpineBase].Position.Y;
1183         float c6 = body.Joints[JointType.SpineBase].Position.Z;
1184
1185         float a7 = body.Joints[JointType.Neck].Position.X;
1186         float b7 = body.Joints[JointType.Neck].Position.Y;
1187         float c7 = body.Joints[JointType.Neck].Position.Z;
1188
1189         float a8 = body.Joints[JointType.Head].Position.X;
1190         float b8 = body.Joints[JointType.Head].Position.Y;
1191         float c8 = body.Joints[JointType.Head].Position.Z;
1192
1193
1194         Vector3 shoulderright = new Vector3(x1, y1, z1);
1195         Vector3 shoulderleft = new Vector3(x2, y2, z2);
1196         Vector3 wristright = new Vector3(a1, b1, c1);
1197         Vector3 wristleft = new Vector3(a2, b2, c2);
1198         Vector3 spinemidv = new Vector3(a3, b3, c3);
1199         Vector3 Elbowright = new Vector3(a4, b4, c4);
1200         Vector3 Elbowleft = new Vector3(a5, b5, c5);
1201         Vector3 SpineB = new Vector3(a6, b6, c6);
1202         Vector3 neck = new Vector3(a7, b7, c7);
1203         Vector3 head = new Vector3(a8, b8, c8);
1204
1205
1206         Vector3 shoulders = new Vector3(x2 - x1, y2 - y1, z2 -
1207         z1);
1208         Vector3 rightarm = new Vector3(a4 - x1, b4 - y1, c4 -
1209         z1);
1210         Vector3 leftarm = new Vector3(a5 - x2, b5 - y2, c5 - z2);
1211
1212         Plane Chest = new Plane();
1213         Chest.Set3Points(shoulderright, shoulderleft, spinemidv);
1214         Vector3 ChestVector = new Vector3();
1215         ChestVector = -Chest.normal;
1216         float HABDUPleft = Vector3.Angle(ChestVector, leftarm);
1217
1218         float HABDUPright = Vector3.Angle(ChestVector, rightarm);
1219
1220         float FLEXleft = Vector3.Angle(-shoulders, leftarm);
1221
1222         float FLEXright = Vector3.Angle(shoulders, rightarm);
1223
1224         StartCoroutine(TemporizadorBOMBA(HABDUPleft, HABDUPright,
1225         FLEXleft, FLEXright, shoulderright, shoulderleft,
1226         wristright, wristleft, spinemidv, Elbowright, Elbowleft,
1227         SpineB, neck, head));
1228     }

```

```

1225 // :::::::::::::: SYMMETRY :::::::::::::: // ↗
1226
1227 if (Input.GetKeyDown(KeyCode.C))
1228 {
1229     //variables
1230     float relbowX = body.Joints ↗
1231     [JointType.ElbowRight].Position.X;
1232     float relbowY = body.Joints ↗
1233     [JointType.ElbowRight].Position.Y;
1234     float relbowZ = body.Joints ↗
1235     [JointType.ElbowRight].Position.Z;
1236     float lelbowX = body.Joints ↗
1237     [JointType.ElbowLeft].Position.X;
1238     float lelbowY = body.Joints ↗
1239     [JointType.ElbowLeft].Position.Y;
1240     float lelbowZ = body.Joints ↗
1241     [JointType.ElbowLeft].Position.Z;
1242     float wristXr = body.Joints ↗
1243     [JointType.WristRight].Position.X;
1244     float wristYr = body.Joints ↗
1245     [JointType.WristRight].Position.Y;
1246     float wristZr = body.Joints ↗
1247     [JointType.WristRight].Position.Z;
1248     float wristXl = body.Joints ↗
1249     [JointType.WristLeft].Position.X;
1250     float wristYl = body.Joints ↗
1251     [JointType.WristLeft].Position.Y;
1252     float wristZl = body.Joints ↗
1253     [JointType.WristLeft].Position.Z;
1254     float shoulderXr = body.Joints ↗
1255     [JointType.ShoulderRight].Position.X;
1256     float shoulderYr = body.Joints ↗
1257     [JointType.ShoulderRight].Position.Y;
1258     float shoulderZr = body.Joints ↗
1259     [JointType.ShoulderRight].Position.Z;
1260     float shoulderXl = body.Joints ↗
1261     [JointType.ShoulderLeft].Position.X;
1262     float shoulderYl = body.Joints ↗
1263     [JointType.ShoulderLeft].Position.Y;
1264     float shoulderZl = body.Joints ↗
1265     [JointType.ShoulderLeft].Position.Z;
1266     float difElbows = Mathf.Abs(relbowY - lelbowY) * 100;
1267     float difWrists = Mathf.Abs(wristYr - wristYl) * 100;
1268     float difShoulders = Math.Abs(shoulderYr - shoulderYl) * ↗
1269     100;
1270
1271     float difElbowsZ = Mathf.Abs(relbowZ - lelbowZ) * 100;
1272     float difWristsZ = Mathf.Abs(wristZr - wristZl) * 100;
1273
1274     Debug.Log("Elbow Right Heigh = " + relbowY + " Elbow ↗
1275     Left Heigh = " + lelbowY);
1276     Debug.Log("Wrist Right Heigh = " + wristYr + " Wrist ↗
1277     Left Heigh = " + wristYl);
1278     Debug.Log("Shoulder Right Heigh = " + shoulderYr + " ↗
1279     Shoulder Left Height = " + shoulderYl);

```

```

1258         Debug.Log("Elbow Right Depth = " + relbowZ + " Elbow Left Depth = " + lelbowZ);
1259         Debug.Log("Wrist Right Depth = " + wristZr + " Wrist Left Depth = " + wristZl);
1260         Debug.Log("Shoulder Right Depth = " + shoulderZr + " Shoulder Left Depth = " + shoulderZl);
1261
1262         Debug.Log("Difference between wrists: " + difWrists);
1263         Debug.Log("Difference between shoulders: " + difShoulders);
1264         Debug.Log("Difference between elbows: " + difElbows);
1265         Debug.Log("hola");
1266         //results
1267
1268
1269         //File results
1270         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Height Difference between wrists: " + difWrists);
1271         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", Environment.NewLine);
1272         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Height Difference between elbows " + difElbows);
1273         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", Environment.NewLine);
1274         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Height Difference between shoulders " + difShoulders);
1275         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", Environment.NewLine);
1276         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Depth Difference between wrists " + difWristsZ);
1277         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", Environment.NewLine);
1278         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Depth Difference between elbows " + difElbowsZ);
1279         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", Environment.NewLine);
1280
1281
1282
1283     }
1284
1285     // :::::::::::::::::::: WRIST EXTENSION :::::::::::::::::::: //
1286
1287     if (Input.GetKeyDown(KeyCode.X))
1288     {
1289         //variables
1290         float wristXr = body.Joints [JointType.WristRight].Position.X;
1291         float wristYr = body.Joints [JointType.WristRight].Position.Y;
1292         float wristZr = body.Joints [JointType.WristRight].Position.Z;
1293         float wristXl = body.Joints [JointType.WristLeft].Position.X;
1294         float wristYl = body.Joints

```

```

1295     [JointType.WristLeft].Position.Y;
1296     float wristZl = body.Joints
1297     [JointType.WristLeft].Position.Z;
1298     float handtipXr = body.Joints
1299     [JointType.HandTipRight].Position.X;
1300     float handtipYr = body.Joints
1301     [JointType.HandTipRight].Position.Y;
1302     float handtipZr = body.Joints
1303     [JointType.HandTipRight].Position.Z;
1304     float handtipXl = body.Joints
1305     [JointType.HandTipLeft].Position.X;
1306     float handtipYl = body.Joints
1307     [JointType.HandTipLeft].Position.Y;
1308     float handtipZl = body.Joints
1309     [JointType.HandTipLeft].Position.Z;
1310     float relbowX = body.Joints
1311     [JointType.ElbowRight].Position.X;
1312     float relbowY = body.Joints
1313     [JointType.ElbowRight].Position.Y;
1314     float relbowZ = body.Joints
1315     [JointType.ElbowRight].Position.Z;
1316     float lelbowX = body.Joints
1317     [JointType.ElbowLeft].Position.X;
1318     float lelbowY = body.Joints
1319     [JointType.ElbowLeft].Position.Y;
1320     float lelbowZ = body.Joints
1321     [JointType.ElbowLeft].Position.Z;
1322
1323     //right side results
1324     Vector3 vecTipandHandr = new Vector3(handtipXr - wristXr,
1325     handtipYr - wristYr, handtipZr - wristZr);
1326     Vector3 vecElbandHandr = new Vector3(relbowX - wristXr,
1327     relbowY - wristYr, relbowZ - wristZr);
1328     float angleWristr = Vector3.Angle(vecTipandHandr,
1329     vecElbandHandr);
1330     Debug.Log("New right wrist extension:" + angleWristr);
1331
1332     //left side results
1333     Vector3 vecTipandHandl = new Vector3(handtipXl - wristXl,
1334     handtipYl - wristYl, handtipZl - wristZl);
1335     Vector3 vecElbandHandl = new Vector3(lelbowX - wristXl,
1336     lelbowY - wristYl, lelbowZ - wristZl);
1337     float angleWristl = Vector3.Angle(vecTipandHandl,
1338     vecElbandHandl);
1339     Debug.Log("New left wrist extension:" + angleWristl);
1340
1341     //File results
1342     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
1343     "Right Wrist Extension: " + angleWristr
1344     + " Left Wrist Extension" + angleWristl);
1345     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
1346     Environment.NewLine);
1347 }
1348
1349 // :::::::::::::::::::::: ABD
1350 HORIZONTAL :::::::::::::::::::::: //

```



```

1328
1329     if (Input.GetKeyDown(KeyCode.D))
1330     {
1331         //variables
1332         float x1 = body.Joints [JointType.ShoulderRight].Position.X;
1333         float y1 = body.Joints [JointType.ShoulderRight].Position.Y;
1334         float z1 = body.Joints [JointType.ShoulderRight].Position.Z;
1335         float x2 = body.Joints [JointType.ShoulderLeft].Position.X;
1336         float y2 = body.Joints [JointType.ShoulderLeft].Position.Y;
1337         float z2 = body.Joints [JointType.ShoulderLeft].Position.Z;
1338         float c1 = body.Joints[JointType.ElbowRight].Position.X;
1339         float c2 = body.Joints[JointType.ElbowRight].Position.Y;
1340         float c3 = body.Joints[JointType.ElbowRight].Position.Z;
1341         float c1l = body.Joints[JointType.ElbowLeft].Position.X;
1342         float c2l = body.Joints[JointType.ElbowLeft].Position.Y;
1343         float c3l = body.Joints[JointType.ElbowLeft].Position.Z;
1344         float n1 = body.Joints[JointType.Neck].Position.X;
1345         float n2 = body.Joints[JointType.Neck].Position.Y;
1346         float n3 = body.Joints[JointType.Neck].Position.Z;
1347
1348         //ref vectors and planes
1349         Vector3 vectorRef = new Vector3(x1 - x2, y1 - y2, z1 -
1350         z2);
1351         Vector3 vecShrNe = new Vector3(x1 - n1, y1 - n2, z1 -
1352         n3);
1353         Vector3 vecShlNe = new Vector3(x2 - n1, y2 - n2, z2 -
1354         n3);
1355         Vector3 plane;
1356         plane = Vector3.Cross(vecShrNe, vecShlNe);
1357         float d1 = plane.x * (-x1) + plane.y * (-y1) + plane.z *
1358         (-z1);
1359
1360         //right side results
1361         float lambda = (((-plane.x) * c1 + (-plane.y) * c2 + (-
1362         plane.z) * c3) - d1) / ((plane.x) * (plane.x) + (plane.y)
1363         * (plane.y) + (plane.z) * (plane.z));
1364         Vector3 elbowrproject = new Vector3(c1 + (lambda *
1365         plane.x), c2 + (lambda * plane.y), c3 + (lambda *
1366         plane.z));
1367         Vector3 vecElShr = new Vector3(elbowrproject.x - x1,
1368         elbowrproject.y - y1, elbowrproject.z - z1);
1369         float angleABDhoriz = Vector3.Angle(vectorRef, vecElShr);
1370         Debug.Log("Angle Right ABD Horiz: " + angleABDhoriz);
1371
1372         //left side results
1373         float lambda2 = (((-plane.x) * c1l + (-plane.y) * c2l +
1374         (-plane.z) * c3l) - d1) / ((plane.x) * (plane.x) +
1375         (plane.y) * (plane.y) + (plane.z) * (plane.z));
1376         Vector3 elbowrprojectl = new Vector3(c1l + (lambda2 *
1377         plane.x), c2l + (lambda2 * plane.y), c3l + (lambda2 *

```

```

plane.z));
1366 Vector3 vecElSh1 = new Vector3(elbowrproject1.x - x2,
elbowrproject1.y - y2, elbowrproject1.z - z2);
1367 float angleABDhoriz1 = Vector3.Angle(vectorRef,
vecElSh1);
1368 Debug.Log("Angle Left ABD Horiz: " + angleABDhoriz1);
1369
1370 //File results
1371 File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
"Right Horizontal ABD: " + angleABDhoriz
+ " Left Horizontal ABD" + angleABDhoriz1);
1372 File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
Environment.NewLine);
1373
1374 }
1375
1376
1377
1378 // :::::::::::::::::::: EXTERNAL
ROTATION :::::::::::::::::::: //
1379
1380 if (Input.GetKeyDown(KeyCode.F))
1381 {
1382     //variables
1383     float x1 = body.Joints
[JointType.ShoulderRight].Position.X;
1384     float y1 = body.Joints
[JointType.ShoulderRight].Position.Y;
1385     float z1 = body.Joints
[JointType.ShoulderRight].Position.Z;
1386     float x2 = body.Joints
[JointType.ShoulderLeft].Position.X;
1387     float y2 = body.Joints
[JointType.ShoulderLeft].Position.Y;
1388     float z2 = body.Joints
[JointType.ShoulderLeft].Position.Z;
1389     float x3 = body.Joints[JointType.SpineMid].Position.X;
1390     float y3 = body.Joints[JointType.SpineMid].Position.Y;
1391     float z3 = body.Joints[JointType.SpineMid].Position.Z;
1392     float u1 = body.Joints[JointType.SpineBase].Position.X;
1393     float u2 = body.Joints[JointType.SpineBase].Position.Y;
1394     float u3 = body.Joints[JointType.SpineBase].Position.Z;
1395     float c1 = body.Joints[JointType.ElbowRight].Position.X;
1396     float c2 = body.Joints[JointType.ElbowRight].Position.Y;
1397     float c3 = body.Joints[JointType.ElbowRight].Position.Z;
1398     float c1l = body.Joints[JointType.ElbowLeft].Position.X;
1399     float c2l = body.Joints[JointType.ElbowLeft].Position.Y;
1400     float c3l = body.Joints[JointType.ElbowLeft].Position.Z;
1401     float w1r = body.Joints[JointType.WristRight].Position.X;
1402     float w2r = body.Joints[JointType.WristRight].Position.Y;
1403     float w3r = body.Joints[JointType.WristRight].Position.Z;
1404     float w1l = body.Joints[JointType.WristLeft].Position.X;
1405     float w2l = body.Joints[JointType.WristLeft].Position.Y;
1406     float w3l = body.Joints[JointType.WristLeft].Position.Z;
1407
1408     //ref vectors and planes
1409     Vector3 vector1 = new Vector3(x2 - x1, y2 - y1, z2 - z1);

```

```

1410     Vector3 vector2 = new Vector3(x3 - x1, y3 - y1, z3 - z1);
1411     Vector3 cp;
1412     cp = Vector3.Cross(vector1, vector2);
1413     Vector3 normalvect = new Vector3(cp.x, cp.y, cp.z);
1414     Vector3 auxvect2 = new Vector3(x3 - u1, y3 - u2, z3 - u3);
1415     Vector3 newcp;
1416     newcp = Vector3.Cross(normalvect, auxvect2);
1417     float daux = newcp.x * (-x3) + newcp.y * (-y3) + newcp.z * (-z3);

1418
1419     //right projections and results
1420     float lambdaaux = (((-newcp.x) * c1 + (-newcp.y) * c2 + (-newcp.z) * c3) - daux) / ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
1421     Vector3 elbowrproject = new Vector3(c1 + (lambdaaux * newcp.x), c2 + (lambdaaux * newcp.y), c3 + (lambdaaux * newcp.z));
1422     float lambdaaux2 = (((-newcp.x) * w1r + (-newcp.y) * w2r + (-newcp.z) * w3r) - daux) / ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
1423     Vector3 wristrproject = new Vector3(w1r + (lambdaaux2 * newcp.x), w2r + (lambdaaux2 * newcp.y), w3r + (lambdaaux2 * newcp.z));
1424     Vector3 vecElbWrisr = new Vector3(elbowrproject.x - wristrproject.x, elbowrproject.y - wristrproject.y, elbowrproject.z - wristrproject.z);
1425     float angleExtRotr = Vector3.Angle(vecElbWrisr, auxvect2);
1426     float angleCorrec = angleExtRotr - 90;
1427     Debug.Log("Angle Ext Rot Right: " + angleCorrec);
1428
1429     //left projections and results
1430     float lambdaaux1 = (((-newcp.x) * c1l + (-newcp.y) * c2l + (-newcp.z) * c3l) - daux) / ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
1431     Vector3 elbowprojectl = new Vector3(c1l + (lambdaaux1 * newcp.x), c2l + (lambdaaux1 * newcp.y), c3l + (lambdaaux1 * newcp.z));
1432     float lambdaaux2l = (((-newcp.x) * w1l + (-newcp.y) * w2l + (-newcp.z) * w3l) - daux) / ((newcp.x) * (newcp.x) + (newcp.y) * (newcp.y) + (newcp.z) * (newcp.z));
1433     Vector3 wristrprojectl = new Vector3(w1l + (lambdaaux2l * newcp.x), w2l + (lambdaaux2l * newcp.y), w3l + (lambdaaux2l * newcp.z));
1434     Vector3 vecElbWrisl = new Vector3(elbowprojectl.x - wristrprojectl.x, elbowprojectl.y - wristrprojectl.y, elbowprojectl.z - wristrprojectl.z);
1435     float angleExtRotl = Vector3.Angle(vecElbWrisl, auxvect2);
1436     float angleCorrecl = angleExtRotl - 90;
1437     Debug.Log("Angle Ext Rot Left: " + angleCorrecl);
1438
1439     //File results
1440     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt", "Right External Rotation: " + angleCorrec

```

```

1441         + " Right External Rotation" + angleCorrec1);
1442     File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
1443         Environment.NewLine);
1444
1445 }
1446
1447 // :::::::::::::::::::: TOUCHING SHOULDERS YER OR
1448 NO :::::::::::::::::::: // se necesita modificar para
1449 tocar la rampa
1450 if (Input.GetKeyDown(KeyCode.G))
1451 {
1452     //variables
1453     float handtipYr = body.Joints
1454     [JointType.HandTipRight].Position.Y;
1455     float handtipZr = body.Joints
1456     [JointType.HandTipRight].Position.Z;
1457     float shoulderYr = body.Joints
1458     [JointType.ShoulderRight].Position.Y;
1459     float shoulderZr = body.Joints
1460     [JointType.ShoulderRight].Position.Z;
1461     float handtipYl = body.Joints
1462     [JointType.HandTipLeft].Position.Y;
1463     float handtipZl = body.Joints
1464     [JointType.HandTipLeft].Position.Z;
1465     float shoulderYl = body.Joints
1466     [JointType.HandTipLeft].Position.Y;
1467     float shoulderZl = body.Joints
1468     [JointType.HandTipLeft].Position.Z;
1469
1470     //differences
1471     float touchYl = Mathf.Abs(shoulderYl - handtipYl) * 100;
1472     float touchZl = Mathf.Abs(shoulderZl - handtipZl) * 100;
1473     float touchYr = Mathf.Abs(shoulderYr - handtipYr) * 100;
1474     float touchZr = Mathf.Abs(shoulderZr - handtipZr) * 100;
1475
1476     //right side results and fileresults
1477     if (touchYr < 5 && touchZr < 15)
1478     {
1479         Debug.Log("Right Hand touch Right Shoulder");
1480         //File.AppendAllText(@"C:\Users\Aleix\Desktop
1481         \Results.txt", "Right Hand Touch Right Shoulder");
1482         //File.AppendAllText(@"C:\Users\Aleix\Desktop
1483         \Results.txt", Environment.NewLine);
1484     }
1485     else
1486     {
1487         Debug.Log("Right Hand don't touch Right Shoulder");
1488         //File.AppendAllText(@"C:\Users\Aleix\Desktop
1489         \Results.txt", "Right Hand Don't Touch Right Shoulder ");
1490         //File.AppendAllText(@"C:\Users\Aleix\Desktop
1491         \Results.txt", Environment.NewLine);
1492     }
1493
1494     //left side results and file results
1495     if (touchYl < 5 && touchZl < 15)

```

UNIVERSITAT POLITÈCNICA DE CATALUNYA 131

```

1522         float neckMove1 = Vector3.Angle(vecHeandNeckr,
vecShoandNeck1);
1523         Debug.Log("Neck movement degrees left part" + neckMove1);
1524
1525
1526         //File results
1527         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
"Angle Head - Right Shoulder : " + neckMover
1528             + " Angle Head - Left Shoulder" + neckMove1);
1529         File.AppendAllText(@"C:\Users\Aleix\Desktop\Results.txt",
Environment.NewLine);
1530
1531
1532     }
1533
1534
1535
1536
1537
1538
1539
1540     }
1541 }
1542
1543 }
1544
1545
1546 private GameObject CreateBodyObject(ulong id)
1547 {
1548     GameObject body = new GameObject("Body:" + id);
1549
1550     for (Kinect.JointType jt = Kinect.JointType.SpineBase; jt <=
Kinect.JointType.ThumbRight; jt++)
1551     {
1552         GameObject jointObj = GameObject.CreatePrimitive
(PrimitiveType.Cube);
1553
1554         LineRenderer lr = jointObj.AddComponent<LineRenderer>();
1555         lr.SetVertexCount(2);
1556         lr.material = BoneMaterial;
1557         lr.SetWidth(0.0f, 0.0f);
1558
1559         jointObj.transform.localScale = new Vector3(0.0f, 0.0f, 0.0f);
1560         jointObj.name = jt.ToString();
1561         jointObj.transform.parent = body.transform;
1562     }
1563
1564     return body;
1565
1566
1567 }
1568
1569 private void RefreshBodyObject(Kinect.Body body, GameObject bodyObject)
1570 {
1571     for (Kinect.JointType jt = Kinect.JointType.SpineBase; jt <=
Kinect.JointType.ThumbRight; jt++)

```

```

1572     {
1573         Kinect.Joint sourceJoint = body.Joints[jt];
1574         Kinect.Joint? targetJoint = null;
1575
1576         if (_BoneMap.ContainsKey(jt))
1577         {
1578             targetJoint = body.Joints[_BoneMap[jt]];
1579         }
1580
1581         Transform jointObj = bodyObject.transform.Find(jt.ToString());
1582         jointObj.localPosition = GetVector3FromJoint(sourceJoint);
1583
1584         LineRenderer lr = jointObj.GetComponent<LineRenderer>();
1585         if (targetJoint.HasValue)
1586         {
1587             lr.SetPosition(0, jointObj.localPosition);
1588             lr.SetPosition(1, GetVector3FromJoint(targetJoint.Value));
1589             lr.SetColors(GetColorForState(sourceJoint.TrackingState),
1590                         GetColorForState(targetJoint.Value.TrackingState));
1591         }
1592         else
1593         {
1594             lr.enabled = false;
1595         }
1596     }
1597
1598     private static Color GetColorForState(Kinect.TrackingState state)
1599     {
1600         switch (state)
1601         {
1602             case Kinect.TrackingState.Tracked:
1603                 return Color.green;
1604
1605             case Kinect.TrackingState.Inferred:
1606                 return Color.red;
1607
1608             default:
1609                 return Color.black;
1610         }
1611     }
1612
1613     private static Vector3 GetVector3FromJoint(Kinect.Joint joint)
1614     {
1615
1616         return new Vector3(joint.Position.X * 10, joint.Position.Y * 10,
1617                             joint.Position.Z * 10);
1618     }
1619
1620
1621
1622
1623
1624
1625

```

**bomba**

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using Kinect = Windows.Kinect;
6 using Windows.Kinect;
7 using UnityEngine.Audio;
10 public class bomba : MonoBehaviour {
11     public Material BoneMaterial;
12     public GameObject BodySourceManager;
13     public GameObject _personaje1_Unity;
14     public GameObject Master;
15     public GameObject Reference;
16     public GameObject Hips;
17     public GameObject LeftUpLeg;
18     public GameObject LeftLeg;
19     public GameObject RightUpLeg;
20     public GameObject RightLeg;
21     public GameObject Spine;
22     public GameObject Spine1;
23     public GameObject Spine2;
24     public GameObject LeftShoulder;
25     public GameObject LeftArm;
26     public GameObject LeftForeArm;
27     public GameObject LeftHand;
28     public GameObject Neck;
29     public GameObject Head;
30     public GameObject RightShoulder;
31     public GameObject RightArm;
32     public GameObject RightForeArm;
33     public GameObject RightHand;
34     public GameObject LeftFoot; //
35     public GameObject RightFoot; //
36     public GameObject RightAnkle; //
37     public GameObject LeftAnkle; //
38     public GameObject pushbuttonL;
39     public GameObject pushbuttonR;
40     public float x;
41     public float y;
42     public float z;
43     public float xbuttonR;
44     public float ybuttons;
45     public float zbuttons;
46     public float xbuttonL;
47     public float yCam;
48     public float zCam;
49     public GameObject Cub;
50     public GameObject Cam;
51     public int numero;
52     public GameObject Bombaa;
53     public GameObject Bomba;
54     public static float Nivell=0;
55     public float countdown = 14f;
56     public Text countdowntext = null;

```



```

57     public Text level = null;
58     public Text levelCountdown = null;
59
60     public GameObject BombExplosion;
61     public bool explotat = false;
62     public GameObject bombSmoke;
63     public GameObject Desactivation;
64     public GameObject CanvasChrono;
65     public GameObject Canvas;
66     public GameObject GameOver;
67     public GameObject Win;
68
69     public GameObject[] DNI;
70     public string nombre;
71     public string dataDia;
72     public GameObject explosion;
73     public GameObject neutralizada_sound;
74     public GameObject explosion_sound;
75     public static int PlayBomb;
76
77     public static float Bombtime;
78
79     // Use this for initialization
80     private Dictionary<Kinect.JointType, Kinect.JointType> _BoneMap = new Dictionary<Kinect.JointType, Kinect.JointType>()
81     {
82         { Kinect.JointType.FootLeft, Kinect.JointType.AnkleLeft },
83         { Kinect.JointType.AnkleLeft, Kinect.JointType.KneeLeft },
84         { Kinect.JointType.KneeLeft, Kinect.JointType.HipLeft },
85         { Kinect.JointType.HipLeft, Kinect.JointType.SpineBase },
86
87         { Kinect.JointType.FootRight, Kinect.JointType.AnkleRight },
88         { Kinect.JointType.AnkleRight, Kinect.JointType.KneeRight },
89         { Kinect.JointType.KneeRight, Kinect.JointType.HipRight },
90         { Kinect.JointType.HipRight, Kinect.JointType.SpineBase },
91
92         { Kinect.JointType.HandTipLeft, Kinect.JointType.HandLeft },
93         { Kinect.JointType.ThumbLeft, Kinect.JointType.HandLeft },
94         { Kinect.JointType.HandLeft, Kinect.JointType.WristLeft },
95         { Kinect.JointType.WristLeft, Kinect.JointType.ElbowLeft },
96         { Kinect.JointType.ElbowLeft, Kinect.JointType.ShoulderLeft },
97         { Kinect.JointType.ShoulderLeft, Kinect.JointType.SpineShoulder },
98
99         { Kinect.JointType.HandTipRight, Kinect.JointType.HandRight },
100        { Kinect.JointType.ThumbRight, Kinect.JointType.HandRight },
101        { Kinect.JointType.HandRight, Kinect.JointType.WristRight },
102        { Kinect.JointType.WristRight, Kinect.JointType.ElbowRight },
103        { Kinect.JointType.ElbowRight, Kinect.JointType.ShoulderRight },
104        { Kinect.JointType.ShoulderRight, Kinect.JointType.SpineShoulder },
105
106        { Kinect.JointType.SpineBase, Kinect.JointType.SpineMid },
107        { Kinect.JointType.SpineMid, Kinect.JointType.SpineShoulder },
108        { Kinect.JointType.SpineShoulder, Kinect.JointType.Neck },
109        { Kinect.JointType.Neck, Kinect.JointType.Head },
110    };
111

```

```

112 void Start () {
113     _personaje1_Unity = GameObject.Find("personaje1_Unity");
114     Master = GameObject.Find("master");
115     Reference = GameObject.Find("Reference");
116     Hips = GameObject.Find("Hips");
117     LeftUpLeg = GameObject.Find("LeftUpLeg");
118     LeftLeg = GameObject.Find("LeftLeg");
119     RightUpLeg = GameObject.Find("RightUpLeg");
120     RightLeg = GameObject.Find("RightLeg");
121     Spine = GameObject.Find("Spine");
122     Spine1 = GameObject.Find("Spine1");
123     Spine2 = GameObject.Find("Spine2");
124     LeftShoulder = GameObject.Find("LeftShoulder");
125     LeftArm = GameObject.Find("LeftArm");
126     LeftForeArm = GameObject.Find("LeftForeArm");
127     LeftHand = GameObject.Find("LeftHand");
128     Neck = GameObject.Find("Neck");
129     Head = GameObject.Find("Head");
130     RightShoulder = GameObject.Find("RightShoulder");
131     RightArm = GameObject.Find("RightArm");
132     RightForeArm = GameObject.Find("RightForeArm");
133     RightHand = GameObject.Find("RightHand");
134     RightAnkle = GameObject.Find("RightToeBase"); //
135     LeftAnkle = GameObject.Find("LeftToeBase"); //
136     RightFoot = GameObject.Find("RightFoot"); //
137     LeftFoot = GameObject.Find("LeftFoot"); //
138     Cub = GameObject.Find("Cube");
139     Cam = GameObject.Find("camara Personaje");
140     pushbuttonL = GameObject.Find("pushbuttonL");
141     pushbuttonR = GameObject.Find("pushbuttonR");
142     Bombaa = GameObject.Find("Bombaa");
143     Bomba = GameObject.Find("Bomba");
144     CanvasChrono = GameObject.Find("CanvasChrono");
145     Canvas = GameObject.Find("Canvas");
146     GameOver = GameObject.Find("GameOver");
147     Win = GameObject.Find("Win");
148     neutralizada_sound = GameObject.Find("neutralizada_sound");
149     explosion_sound = GameObject.Find("explosion_sound");
150
151     GameOver.SetActive(false);
152     Win.SetActive(false);
153 }
154
155 IEnumerator Temporizador()
156 {
157     yield return new WaitForSeconds(3);
158     Nivell = Nivell + 1;
159     pushbuttonR.gameObject.active = true;
160     pushbuttonL.gameObject.active = true;
161     Bombaa.gameObject.active = true;
162     countdown = 15;
163 }
164
165 IEnumerator TempGameOver()
166 {
167     yield return new WaitForSeconds(2);

```

```

168    GameOver.SetActive(true);
169 }
170
171 // Update is called once per frame
172 void Explode()
173 {
174     Bombaa.gameObject.active = false;
175     BombExplosion = GameObject.Find("BombExplosion");
176     var exp = BombExplosion.GetComponent<ParticleSystem>();
177     exp.Play();
178     explotat = true;
179     explosion_sound.SendMessage("Boom");
180     StartCoroutine(TempGameOver());
181     PlayBomb = 0;
182     pushbuttonR.SetActive(false);
183     pushbuttonL.SetActive(false);
184 }
185
186 void BombDesactivation()
187 {
188     Desactivation = GameObject.Find("Desactivation");
189     var exp1 = Desactivation.GetComponent<ParticleSystem>();
190     exp1.Play();
191     neutralizada_sound.SendMessage("BombOut");
192 }
193
194 void Update () {
195     Bombtime += Time.deltaTime;
196
197     if (!pushbuttonR.gameObject.active && !pushbuttonL.gameObject.active)
198     {
199         countdown = 99;
200     }
201
202     if (countdown >= 9.5f)
203     {
204         countdowntext.text = ("00 : " + countdown.ToString("f0"));
205     }
206
207     if (countdown < 9.5f)
208     {
209         countdowntext.text = ("00 : 0" + countdown.ToString("f0"));
210     }
211     if (countdown < 3f && explotat==false)
212     {
213
214         //BombSmoke();
215     }
216
217     if (countdown <= 0)
218     {
219         countdown = 0;
220         if (explotat ==false)
221         { Explode();
222           explotat = true;
223         }

```

```
224     }
225
226     if (Nivell == 0)
227     {
228
229         level.text = "1";
230         ybuttons = y*1.0f;
231         CanvasChrono.SetActive(false);
232         Canvas.SetActive(true);
233
234     }
235     if (Nivell == 1)
236     {
237         CanvasChrono.SetActive(true);
238         Canvas.SetActive(false);
239         levelCountdown.text = ("2");
240
241         countdown -= Time.deltaTime;
242         ybuttons = y * 1.0f;
243     }
244
245     if (Nivell == 2)
246     {
247         CanvasChrono.SetActive(false);
248         Canvas.SetActive(true);
249         level.text = ("3");
250         ybuttons = y * 1.2f;
251
252     }
253
254     if (Nivell == 3)
255     {
256         CanvasChrono.SetActive(true);
257         Canvas.SetActive(false);
258
259         levelCountdown.text = ("4");
260         countdown -= Time.deltaTime;
261         ybuttons = y * 1.2f;
262
263     }
264
265     if (Nivell == 4)
266     {
267         CanvasChrono.SetActive(false);
268         Canvas.SetActive(true);
269         level.text = ("5");
270         ybuttons = y * 1.3f;
271
272     }
273     if (Nivell == 5)
274     {
275         CanvasChrono.SetActive(true);
276         Canvas.SetActive(false);
277         levelCountdown.text = ("6");
278
279         countdown -= Time.deltaTime;
```

```

280         ybuttons = y * 1.3f;
281
282     }
283     if (Nivell == 6)
284     {
285         CanvasChrono.SetActive(false);
286         Canvas.SetActive(true);
287         level.text = ("7");
288         ybuttons = y * 1.4f;
289
290     }
291     if (Nivell == 7)
292     {
293         CanvasChrono.SetActive(true);
294         Canvas.SetActive(false);
295         levelCountdown.text = ("8");
296
297         countdown -= Time.deltaTime;
298         ybuttons = y * 1.4f;
299
300     }
301
302     if (Nivell == 8)
303     {
304         Win.SetActive(true);
305     }
306
307     x = Head.transform.position.x;
308     y = Head.transform.position.y;
309     z = Head.transform.position.z;
310     xbuttonR = x + 0.4f;
311     xbuttonL = x - 0.4f;
312     zbuttons = z + 0.2f ;
313     yCam = y + 1;
314     zCam = z - 4f;
315
316     pushbuttonL.transform.position = new Vector3(xbuttonL, ybuttons,  ↗
        zbuttons);
317     pushbuttonR.transform.position = new Vector3(xbuttonR, ybuttons,  ↗
        zbuttons);
318     Cam.transform.position = new Vector3(x, yCam,zCam);
319
320     if (pushbutton.numero == 2)
321     {
322         StartCoroutine(Temporizador());
323
324         BombDesactivation();
325         pushbutton.numero = 0;
326     }
327 }
328 }
329

```

**Button\_sound**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class button_sound : MonoBehaviour
6 {
7     private AudioSource _audioSource;
8
9     void Button()
10    {
11        _audioSource = gameObject.GetComponent<AudioSource>();
12
13        _audioSource.Play();
14    }
15 }
16
17
```

**canviaEscena**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class canviaEscena : MonoBehaviour{
7
8     public void canviaBomba()
9     {
10         SceneManager.LoadScene("Joc Bomba");
11     }
12
13     public void backMenu()
14     {
15         SceneManager.LoadScene(3);
16     }
17
18     public void playbomb()
19     {
20         bomba.PlayBomb = 1;
21         bomba.Bombtime = 0;
22     }
23
24     public void quitbomb()
25     {
26         bomba.PlayBomb = 0;
27         bomba.Nivell = 0;
28     }
29 }
30
```

**CopiaDNI**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 /// <summary>
6 /// En la escena Pacientes, en el Panel. Busca el GameObject con el tag "DNI" del doctor, que corresponde al DNI que se coge con DontDestroyOnLoad. Permite que aparezca el DNI del doctor en la escena Pacientes arriba a al izq.
7 /// </summary>
8 public class CopiaDNI : MonoBehaviour {
9     public Text mytext=null;
10    public GameObject[] DNI;
11    public string nombre;
12
13    // Update is called once per frame
14    void Update () {
15        DNI = GameObject.FindGameObjectsWithTag("DNI");
16        nombre = DNI[0].name;
17        mytext.text = nombre;
18    }
19 }
20
```



**DataInsenter**

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4
5
6  /// Registro de Doctores, para pasar todos los datos insertados al servidor  ➤
   (mediante el script IsertUser.php)
7
8  //En la escena GlobalAccountSystem, ir al Register y cada InputField  ➤
   corresponde a cada GameObject de los que vienen a continuación.
9  //Se define el texto asociado a cada GameObject con el "private Text".
10 public class DataInsenter : MonoBehaviour {
11     public GameObject fiiirstname;
12     private Text FirstName;
13     public GameObject LaastName;
14     private Text LastName;
15     // public GameObject Aage;
16     // private Text Age;
17     // public GameObject Coompany;
18     // private Text Company;
19     public GameObject Useername;
20     private Text UserName;
21     // public GameObject Email;
22     // private Text Email;
23     public GameObject Paassword;
24     private Text Password;
25     //Mensaje registrado
26     public Text Mensaje = null;
27
28
29     string CreateUserURL = "localhost/users/InsertUser.php";//Abre el script  ➤
       InsertUser.php
30
31     // Se asigna el texto asociado a cada GameObject.
32     void Start () {
33         FirstName= fiiirstname.GetComponent <Text> ();
34         LastName = LaastName.GetComponent<Text>();
35         // Age = Aage.GetComponent<Text>();
36         // Company = Coompany.GetComponent<Text>();
37         UserName = Useername.GetComponent<Text>();
38         // Email = Email.GetComponent<Text>();
39         Password = Paassword.GetComponent<Text>(); 40
       }
42     // Al pulsar el Boton Register, inicia la funcion CreateUser
44     public void ButtonClick()
45     {
46         StartCoroutine(CreateUser(FirstName.text, LastName.text,UserName.text, ➤
           Password.text));
47     }
48     //Funcion CreateUser: envía los datos de registro al php, que luego los  ➤
       envía al servidor.
49     IEnumerator CreateUser(string FirstName, string LastName, string  ➤
       UserName,string Password)
50     {

```

```
51     WWWForm form = new WWWForm();
52
53     form.AddField("FirstNamePost", FirstName);
54     form.AddField("LastNamePost", LastName);
55     form.AddField("UserNamePost", UserName);
56     form.AddField("PasswordPost", Password);
57
58     WWW www = new WWW(CreateUserURL, form);
59     yield return www;
60
61     Debug.Log(www.text);
62     Mensaje.text = www.text;
63 }
64 }
65
```

**DeletePatient**

```

1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.UI;
4 using UnityEngine.SceneManagement;
5 using System.Collections.Generic; //
6 using System;
7 using System.Text;
8 using UnityEngine.EventSystems;
9 using System.Text.RegularExpressions;
10 using System.Linq;
11 //Escena Pacientes, boton EliminarPaciente. Sirve para coger los datos del
    paciente para eliminar y enviarlos al php.
12 public class DeletePatient : MonoBehaviour
13 {
14     public GameObject PatientNames;
15     public Text PatientName;
16     private string DoctoresName;
17     public GameObject DDNI;
18     private Text DNI;
19     public Button myButton; //boton Refresh //Dado que no puede Eliminar y
        Refrescar pacientes a la vez se le pone un pequeño temporizador de 0.2
        Seg
20
21     string DeletePatientURL = "localhost/users/DeletePatient.php";
22
23     // Use this for initialization
24     void Start()
25     {
26
27         DNI = DDNI.GetComponent<Text>();
28     }
29     public void Update()
30     {
31         PatientName = PatientNames.GetComponent<Text>();
32         DoctoresName = DNI.text;
33     }
34     public void ButtonClick()
35     {
36         StartCoroutine(Lista1(PatientName.text, DoctoresName));
37     }
38
39     IEnumerator Lista1(string PatientName, string DoctoresName)
40     {
41         WWWForm form = new WWWForm();
42
43         form.AddField("PatientNamePost", PatientName);
44         form.AddField("DoctorNamePost", DoctoresName);
45         WWW www = new WWW(DeletePatientURL, form);
46
47         yield return www;
48         //Debug.Log(www.text);
49         StartCoroutine(Temporizador()); //llama a la corutina Temporizador
50
51     }

```

52      `IEnumerator` Temporizador()//Dado que no puede Eliminar yRefrescar  
          pacientes a la vez se le pone un pequeño temporizador de 0.2 Seg



```
53     {  
54         yield return new WaitForSeconds(0.2f);  
55         myButton.onClick.Invoke();  
56     }  
57 }  
58
```

**DestroyOnload**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DestroyOnload : MonoBehaviour{
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14
15    }
16    public void ButtonClick()
17    {
18        GameObject[] names = GameObject.FindGameObjectsWithTag("DNI");
19
20        foreach (GameObject item in names)
21        {
22            Destroy(item);
23        }
24    }
25
26 }
27
```

## Envioangulos

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System.Linq;
5 using System.Text;
6 using UnityEngine.UI;
7
8 public class Envioangulo : MonoBehaviour {
9
10     public string angulo = null;
11     public string angulo2 = null;
12     public Text angulo3 = null;
13     public Text angulo4 = null;
14     public GameObject[] DNI;
15     public string nombre;
16     // prueba envio angulo actual
17
18     //public Text guadarAD1;
19     //public Text guadarAIz;
20     //public Text guardar2;
21     public BodySourceUpperBody bS;
22
23     public Text mytext=null;
24     public string fecha;
25     public float hora;
26     public float minutos;
27     public string Hora;
28     public bool onetime = false;
29     public string Movimiento=null ;
30     public string Simetrianum;
31     public string Nivell;
32     public string Bombtime;
33     public string repHABDUP;
34     public string HABDUptime;
35     public float guai = 0;
36
37
38
39     // Update is called once per frame
40     string AngleInsertURL = "localhost/users/AngleInsert.php";//Abre el script ➤
41     AngleInsert.php
42
43     public void envioHABDUP()
44     {
45         System.DateTime dateTime = System.DateTime.Now;
46         fecha = dateTime.ToString("dd-MM-yyyy");
47         //hora
48         hora = System.DateTime.Now.Hour;
49         minutos = System.DateTime.Now.Minute;
50         Hora = hora + ":" + minutos;
51         //DNI
52
53         DNI = GameObject.FindGameObjectsWithTag("DNI");
54         nombre = DNI[1].name;
55

```

```

56     Movimiento = "HABDUP";
57     angulo = "-";
58     angulo2= "-";
59     Simetrianum = "-";
60
61     Nivell = "-";
62     Bombtime = "-";
63
64     repHABDUP= BodySourceUpperBody.repHABDUP;
65
66     HABDUptime = BodySourceUpperBody.HABDUptime.ToString();//
67     bS.HABDUptime.ToString();
68     StartCoroutine(AngleInsert(nombre, angulo, angulo2, fecha, Hora,
69     Movimiento, Simetrianum, repHABDUP, HABDUptime, Nivell, Bombtime));
70 }
71 public void envioBOMBA()
72 {
73     Debug.Log("chachi");
74
75     System.DateTime dateTime = System.DateTime.Now;
76     fecha = dateTime.ToString("dd-MM-yyyy");
77     //hora
78     hora = System.DateTime.Now.Hour;
79     minutos = System.DateTime.Now.Minute;
80     Hora = hora + ":" + minutos;
81     //DNI
82
83     DNI = GameObject.FindGameObjectsWithTag("DNI");
84     nombre = DNI[1].name;
85
86     Movimiento = "BOMBA";
87     angulo = "-";
88     angulo2 = "-";
89     Simetrianum = "-";
90
91     Nivell = bomba.Nivell.ToString();//(bomba.Nivell.ToString());
92     Bombtime =bomba.Bombtime.ToString();//bomba.Bombtime.ToString();
93
94     repHABDUP = "-";
95
96     HABDUptime = "-"; ;//bS.HABDUptime.ToString();
97     StartCoroutine(AngleInsert(nombre, angulo, angulo2, fecha, Hora,
98     Movimiento, Simetrianum, repHABDUP, HABDUptime, Nivell, Bombtime));
99 }
100 void Update()
101 {
102     // Debug.Log(angulo.text);
103     //Debug.Log(angulo2.text);
104     Movimiento = "Abduction";
105     //Debug.Log(bS.cont);
106     if (bS.cont == 5 && onetime==false)
107     {
108         //fecha

```



```

109         System.DateTime dateTime = System.DateTime.Now;
110         fecha = dateTime.ToString("dd-MM-yyyy");
111         //hora
112         hora = System.DateTime.Now.Hour;
113         minutos = System.DateTime.Now.Minute;
114         Hora = hora + ":" + minutos;
115         //DNI
116
117         DNI = GameObject.FindGameObjectsWithTag("DNI");
118         nombre = DNI[1].name;
119
120         string Nivell = "-";
121         string Bombtime = "-";
122         string RepHABDUP = "-";
123         string HABDUptime = "-";
124         angulo = bS.angulo.text.ToString();
125         angulo2 = bS.angulo2.text.ToString();
126
127         Simetrianum = bS.Simetrianum.text.ToString();
128         StartCoroutine(AngleInsert(nombre, angulo, angulo2, fecha, Hora,
129             Movimiento, Simetrianum, RepHABDUP, HABDUptime, Nivell,
130             Bombtime));
131     }
132
133
134     //mytext.text = nombre;
135     //Debug.Log(nombre);
136
137
138 }
139
140
141
142 IEnumerator AngleInsert(string nombre, string angulo, string angulo2,
143     string fecha, string Hora, string Movimiento, string Simetrianum, string
144     RepHABDUP, string HABDUptime, string Nivell, string Bombtime) //,
145     string guardar2
146 {
147     WWWForm form = new WWWForm();
148
149     form.AddField("DNIPost", nombre);
150     form.AddField("AnguloMaxPost", angulo);
151     form.AddField("AnguloMax2Post", angulo2);
152     form.AddField("FechaPost", fecha);
153     form.AddField("HoraPost", Hora);
154     form.AddField("MovimientoPost", Movimiento);
155     form.AddField("SimetriaPost", Simetrianum);
156     form.AddField("RepHABDUPPost", RepHABDUP);
157     form.AddField("HABDUptimePost", HABDUptime);
158     form.AddField("NivellPost", Nivell);
159     form.AddField("BombtimePost", Bombtime);

```

```
160         WWW www = new WWW(AngleInsertURL, form);
161         yield return www;
162
163
164         //Mensaje.text = www.text;
165     }
166
167
168 }
169
```

**explosion**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class explosion : MonoBehaviour
6 {
7
8     private AudioSource _audioSource;
9
10    void Boom()
11    {
12        _audioSource = gameObject.GetComponent<AudioSource>();
13
14        _audioSource.Play();
15    }
16 }
17
18
19
```

## Listado

```

1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.UI;
4 using UnityEngine.SceneManagement;
5 using System.Collections.Generic; //
6 using System;
7 using System.Text;
8 using UnityEngine.EventSystems;
9 using System.Text.RegularExpressions;
10 using System.Linq;
11
12 /// <summary>
13 /// Se encuentra en la escena Patients-> en el Content. Pide al servidor con el ↗
14 /// SeleccionDoctor.php que le de todos los pacientes de un Doctor (DNI doctor) ↗
15 /// especifico,
16 /// despues hace un split de la lista de pacientes
17 ///
18 /// </summary>
19
20 public class Listado : MonoBehaviour
21 {
22     public GameObject DDNI;
23     private Text DNI;
24     public Text mytext = null;
25     public String[] Pacientes;
26
27     string LoginURL = "localhost/users/SeleccionDoctor.php";
28
29     // Use this for initialization
30     void Start()
31     {
32         DNI = DDNI.GetComponent<Text>();//DNI del doctor
33     }
34
35     void Update()
36     {
37         StartCoroutine(Lista1(DNI.text));
38     }
39
40     IEnumerator Lista1(string DNI)
41     {
42         WWWForm form = new WWWForm();
43         form.AddField("DNIPost", DNI); //Envia el DNI del Dr al servidor
44
45         WWW www = new WWW(LoginURL, form);
46
47         yield return www;
48
49         var Clientes = www.text.Split(new string[] { "\n" }, ↗
50             StringSplitOptions.None); //Separa el mensaje en cada salto de linea
51
52         Pacientes = Clientes;
53         Debug.Log(Pacientes[0]);
54     }
55 }

```

## Login

```

1  //LIBRERIAS
2  using System.Collections;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.SceneManagement;
6
7  /// <summary>
8  /// Script que permite el Login de un Usuario (al darle al botón Login en la  ?
   escaena GlobalAccountSystem): poniendo los datos del Usuario (DNI y  ?
   contraseña),
9  /// se entra en el Perfil del usuario (mediante el script Login.php) y se pasa  ?
   a la siguiente escena "Patients" para poder insertar pacientes a ese Usuario
10 /// </summary>
11 public class Login : MonoBehaviour
12 {
13     //DECLARACIÓN DE VARIABLES
14
15     public GameObject UserName;
16     private Text UserName;
17     public GameObject Paassword;
18     private Text Password;
19
20     public Text Mensaje=null;
21
22     string LoginURL = "localhost/users/Login.php";
23
24     // Use this for initialization
25     void Start()
26     {
27         //TRANSFORMACIN VARIABLE (GameObject a Text)
28         UserName = UserName.GetComponent<Text>();
29         Password = Paassword.GetComponent<Text>(); 30
30     }
31
32     public void ButtonClick()
33     {
34         //INICIO CORUTINA
35         StartCoroutine(LoginToDB(UserName.text, Password.text)); 36
36     }
37
38     IEnumerator LoginToDB(string UserName, string Password)
39     {
40         WWWForm form = new WWWForm();
41
42         form.AddField("UserNamePost", UserName);
43         form.AddField("PasswordPost", Password);
44
45         WWW www = new WWW(LoginURL, form);
46         yield return www;
47         Debug.Log(www.text);
48         //Devuelve los datos recibidos del PHP
49         Mensaje.text= www.text;
50         //Si recibe la frase "Inicio de sesión correcto" procede a iniciar  ?
   sesión

```

```
53         if ( www.text == "Inicio de sesión correcto")
54         {
55             SceneManager.LoadScene("Pacients", LoadSceneMode.Single);
56         }
57     }
58 }
```

## LoginPaciente

```

1  using System.Collections;
2  using UnityEngine;
3  using UnityEngine.UI;
4  using UnityEngine.SceneManagement;
5
6  /// <summary>
7  /// Script que permite el Login de un Paciente (al darle al botón Login en la  ↗
   escena GlobalAccountSystem): poniendo los datos del Paciente (DNI),
8  /// se entra en el Perfil del paciente (mediante el script LoginPaciente.php) y  ↗
   se pasa a la siguiente escena "Movement" para poder escoger los ejercicios
   para hacer.
9  /// </summary>
10
11 public class LoginPaciente : MonoBehaviour {
12
13     public GameObject UserUserName; //realmente es el DNIpaciente
14     private Text UserName;
15     public Text Mensaje = null;
16
17
18     string LoginURL = "localhost/users/LoginPaciente.php";
19
20     // Use this for initialization
21     void Start()
22     {
23         UserName = UserUserName.GetComponent<Text>();
24     }
25
26     public void ButtonClick()
27     {
28         StartCoroutine(LoginToDB(UserName.text));
29     }
30
31     IEnumerator LoginToDB(string UserName)
32     {
33         WWWForm form = new WWWForm();
34         form.AddField("DNIPost", UserName);
35         WWW www = new WWW(LoginURL, form); //envia el DNI al script  ↗
           LoginPaciente.php
36
37         yield return www;
38
39         Debug.Log(www.text);
40         Mensaje.text = www.text;
41         if (www.text == "Inicio de sesión correcto") //comprueba si el DNI  ↗
           existe
42         {
43             //SceneManager.LoadScene("Movement", LoadSceneMode.Single); //se  ↗
               carga la siguiente escena (escena Movement)
44             SceneManager.LoadScene(3);
45         }
46     }
47 }

```

**mostrardias**

```

1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.UI;
4 using UnityEngine.SceneManagement;
5 using System.Collections.Generic; //
6 using System;
7 using System.Text;
8 using UnityEngine.EventSystems;
9 using System.Text.RegularExpressions;
10 using System.Linq;
11
12 public class mostrardias : MonoBehaviour {
13
14     public GameObject noombre;
15     private Text nombres;
16     public Text mytext = null;
17     public String[] Pacientes;
18     public Text pacientee;
19
20     public string paciente;
21
22     string LoginURL = "localhost/users/mostrardias.php";
23
24     void Update()
25     {
26         nombres = noombre.GetComponent<Text>(); //DNI del doctor
27         paciente = nombres.text;
28
29         string paaciente= paciente.Remove(0, 1);
30         //Debug.Log(paaciente);
31         // var bytes = System.Text.Encoding.UTF8.GetBytes(paaciente);
32         //Debug.Log(bytes[0]);
33
34         StartCoroutine(Lista1(paaciente));
35     }
36
37     IEnumerator Lista1(string paaciente)
38     {
39         WWWForm form = new WWWForm();
40
41         form.AddField("PatientNamePost", paaciente); //Envia el DNI del Dr al
            servidor
42
43         //Debug.Log(paaciente);
44         WWW www = new WWW(LoginURL, form);
45
46         yield return www;
47         // Debug.Log(www.text);
48
49         var Dias = www.text.Split(new string[] { "\n" },
            StringSplitOptions.None); //Separa el mensaje en cada salto de linea
50
51         Pacientes = Dias;
52     }
53 }

```



## NameController

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 //Escena Global Account System,GameObject con mismo nombre, coje el DNI del login del doctor y no lo destruye al pasar de escena,para despues tener el DNI presente en la otra escena
5 public class NameController : MonoBehaviour {
6     public GameObject UserrrName;
7     private Text UserName;
8     // Use this for initialization
9
10    void Update () {
11        UserName = UserrrName.GetComponent<Text>();
12        gameObject.name = UserName.text;
13    }
14
15    //No lo destruye en el cambio de escena
16    void Awake () {
17        DontDestroyOnLoad(gameObject);
18        DontDestroyOnLoad(UserName);
19    }
20 }
21 }
22 }
```

**neutralizada**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class neutralizada : MonoBehaviour
6 {
7
8     private AudioSource _audioSource;
9
10     void BombOut()
11     {
12         _audioSource = gameObject.GetComponent<AudioSource>();
13
14         _audioSource.Play();
15     }
16
17 }
18
```

## PatientInsert

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4
5  /// Registro de Pacientes, para pasar todos los datos insertados al servidor  ?
   (mediante el script InsertUser.php)
6  ///
7  //En la escena Pacients, ir al Register y cada InputField corresponde a cada  ?
   GameObject de los que vienen a continuación.
8  //Se define el texto asociado a cada GameObject con el "private Text".
9  public class PatientInsert: MonoBehaviour
10 {
11     public GameObject PaatientName;
12     private Text PatientName;
13     public GameObject PaatientDNI;
14     private Text PatientDNI;
15     public GameObject DDNI;
16     private Text DNI;
17     public Text Mensaje = null;
18     public InputField PacientsName;
19     public InputField DNIi;
20
21     string CreatePatientURL = "localhost/users/InsertPatient.php";//Abre el  ?
       script InsertPatient.php
22
23     // Se asigna el texto asociado a cada GameObject.
24     void Start()
25     {
26         PatientName = PaatientName.GetComponent<Text>();
27         PatientDNI = PaatientDNI.GetComponent<Text>();
28         DNI = DDNI.GetComponent<Text>(); 29
       }
29
30     public void ButtonClick()
31     {
32         CreatePatient(PatientName.text, PatientDNI.text,DNI.text);
33         PacientsName.Select();
34         PacientsName.text = "";
35         DNIi.Select();
36         DNIi.text = ""; 38
       }
37
38     //Funcion CreatePatient: envía los datos de registro al InsertPacient.php,  ?
       que luego los envía al servidor.
39
40     public void CreatePatient(string PatientName, string PatientDNI, string  ?
       DNI)
41     {
42         WWWForm form = new WWWForm();
43
44         form.AddField("PatientNamePost", PatientName);
45         form.AddField("PatientDNIPost", PatientDNI);
46         form.AddField("DNIPost", DNI);
47         Debug.Log(PatientDNI + PatientName + DNI + "");
48
49         WWW www = new WWW(CreatePatientURL, form);
50         Mensaje.text = "Usuario Registrado Correctamente";
51     }

```

**pushbutton**

```

1  using UnityEngine;
2  using System.Collections;
3
4
5  public class pushbutton : MonoBehaviour
6  {
7      public GameObject pushbuttonR;
8      public GameObject pushbuttonL;
9
10     public GameObject deactivate;
11     public GameObject pushbutton_R;
12
13     public static int numero;
14     void Start()
15     {
16         numero = 0;
17         pushbuttonR = GameObject.Find("pushbuttonR");
18         pushbuttonL = GameObject.Find("pushbuttonL");
19         deactivate = GameObject.Find("deactivate");
20         pushbutton_R = GameObject.Find("pushbutton_R");
21         //pushbutton_L = GameObject.Find("pushbutton_L");
22
23     }
24
25     void Desactivate()
26     {
27         var exp = deactivate.GetComponent<ParticleSystem>();
28         exp.Play();
29     }
30
31     void OnCollisionEnter(Collision other)
32     {
33
34         //Destroy(this.gameObject);
35         pushbutton_R.SendMessage("Button");
36         numero = numero + 1;
37         deactivate.transform.position = this.gameObject.transform.position;
38         Desactivate();
39         this.gameObject.active = false;
40     }
41 }
42

```

**RefreshRemoto**

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4  using System.Collections.Generic;
5
6
7  //Dentro Main camera, Escena Pacients
8  //Funcion: El back button del register invoca a este script que invoca al boton ↗
9  // refresh.
10 //Ek boton refresh llama al ShopScroll list que esta dentro del panel "content ↗
11 //Esto permite que al darle al boton back haga un refresh automaticamente de ↗
12 // los pacientes
13 public class RefreshRemoto : MonoBehaviour
14 {
15     public Button myButton; //boton Refresh
16
17     public void ButtonClick()
18     {
19         StartCoroutine(Temporizador());
20     }
21
22     IEnumerator Temporizador() //Dado que no puede Eliminar y Refrescar ↗
23     // pacientes a la vez se le pone un pequeño temporizador de 0.2 Seg
24     {
25         yield return new WaitForSeconds(0.2f);
26         myButton.onClick.Invoke();
27     }
28 }

```

**SalirScena**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 public class SalirScena : MonoBehaviour {
6     //En escena Pacients, Scrollview/canvas/CerrarSesion
7     //Permite, al pulsar el boton, cerrar sesion y volver a la escea inicial
8     public void ButtonClick()
9     {
10         SceneManager.LoadScene("GlobalAccountSystem", LoadSceneMode.Single);
11     }
12 }
13
```

## SampleButton

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4  using System.Collections.Generic;
5  //escena Pacientes. Ubicada dentro el prefab (SampleButton),Funcion:Util para  ↗
   que en el Script shopScroll list pueda aparecer el boton tantas veces como  ↗
   pacientes haya
6  public class SampleButton : MonoBehaviour
7  {
8
9      public Button buttonComponent;
10     public Text nameLabel;
11
12     private Item item;
13     private ShopScrollList scrollList;
14     public GameObject panel;
15     public Text NameText;
16     public Text NameanguloText;
17
18     public void Setup(Item currentItem, ShopScrollList currentScrollList)
19     {
20         item = currentItem;
21         nameLabel.text = item.itemName;//cambiar aqui el nombre del boton
22
23         scrollList = currentScrollList;
24
25     }
26     public void openpanel()
27     {
28         if (panel != null)
29         {
30             bool isActive = panel.activeSelf;
31             panel.SetActive(!isActive); 32
32         }
33     }
34
35     public void ButtonClick()
36     {
37         Debug.Log("copaido");
38
39         NameText.text = this.name;//
40         NameanguloText.text = this.name;
41         Debug.Log(NameanguloText.text); 42
42     }
43
44     }

```

## ShopScrollList

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4  using System.Collections.Generic;
5
6  //Escena Pacients, en Content. Sirve para crear una lista de botones segun el  ➤
   numero de pacientes que haya para un doctor determinado (que se obtiene con  ➤
   el script CopiaDNI, la parte de: GameObject.FindGameObjectsWithTag("DNI"));
7  [System.Serializable]
8  public class Item
9  {
10     public string itemName;
11 }
12
13 public class ShopScrollList : MonoBehaviour
14 {
15     public List<Item> itemList;
16
17     public Transform contentPanel;
18     //public Transform contentPanelEliminar;
19     public SimpleObjectPool buttonObjectPool;
20     // public SimpleObjectPool buttonObjectPool2;
21
22     public string[] PacientesB;
23     public bool refresh;
24
25     public Text mytext = null;
26     public GameObject[] DNI;
27     public string nombre;
28
29     private void Start()
30     {
31         StartCoroutine(Temporizador()); //llama a la corutina Temporizador al  ➤
           iniciar la escena
32     }
33
34     public void ButtonClick()
35     {
36         RemoveButtons();
37
38         Listado variable = GetComponent<Listado>(); //en rojo pero funciona
39         PacientesB = variable.Pacientes;
40
41         for (int i = 0; i < PacientesB.Length - 1; i++) //Bucle For: crea  ➤
           tantos botones como splits "/n" (saltos de linea) devuelva el script  ➤
           Listado -1
42     {
43         49
44         50
45
46
47
48

```



```

// Item item
=
PacientesB[i
];
GameObject
t
newButton
=
buttonObj
ectPool.G
etObject(
);
newButton
.transfor
m.SetPare
nt(conten
tPanel);/
/se crean
los
botones
en el
panel
"Content"

```

```

SampleButt
on
sampleButt
on =
newButton.
GetComponent<SampleB
utton>();
sampleButt
on.GetComponentInChi
ldren<Text
>().text =
PacientesB
[i];
sampleButt
on.name =
PacientesB
[i]; //copi
a el
nombre de
los splits

```

```

51         "/n" (saltos de linea) encima de cada botón
52     /*
53     GameObject newButton2 = buttonObjectPool2.GetObject();
54     newButton2.transform.SetParent(contentPanelEliminar);
55     newButton2.name = PacientesB[i];*/
56 }
57 void RemoveButtons() //para actualizar el numero de botones, se eliminan  P
    todos y se vuelven a crear al darle al botón Refresh de la escena  P
    Pacients.
58 {
59     while (contentPanel.childCount > 0)
60     {
61         GameObject toRemove = contentPanel.transform.GetChild  P
            (0).gameObject;
62         buttonObjectPool1.ReturnObject(toRemove);
63         //GameObject toRemove2 = contentPanelEliminar.transform.GetChild  P
            (0).gameObject;
64         //buttonObjectPool2.ReturnObject(toRemove2);
65     }
66 }
67
68 IEnumerator Temporizador()//Permite que a la iniciar la escena salga  P
    automaticamente la lista de pacientes sin darle al boton refresh despues  P
    de 0.5S
69 {
70     yield return new WaitForSeconds(0.5f);
71     ButtonClick();
72 }
73 }

```

**ShopScrollListDias**

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4  using System.Collections.Generic;
5
6  //
7  [System.Serializable]
8  public class Item2
9  {
10     public string itemName;
11 }
12 public class ShopScrollListDias : MonoBehaviour {
13
14     public List<Item2> itemList;
15
16     public Transform contentPanel;
17     //public Transform contentPanelEliminar;
18     public SimpleObjectPool buttonObjectPool;
19     // public SimpleObjectPool buttonObjectPool2;
20
21     public string[] Dias;
22     public bool refresh;
23
24     public Text mytext = null;
25     public GameObject[] DNI;
26     public string nombre;
27
28     public void Update()
29     {
30         ButtonClick();
31     }
32     private void Start()
33     {
34         StartCoroutine(Temporizador()); //llama a la corutina Temporizador al
35         iniciar la escena
36     }
37     public void ButtonClick()
38     {
39         RemoveButtons();
40
41         mostrardias variable = GetComponent<mostrardias>(); //en rojo pero
42         funciona
43         Dias = variable.Pacientes;
44
45         for (int i = 0; i < Dias.Length - 1; i++) //Bucle For: crea tantos
46             botones como splits "/n" (saltos de linea) devuelva el script Listado
47             -1
48             {
49                 50
50                 51

```

```
// Item item
=
PacientesB[i
];
GameObjec
t
newButton
=
buttonObj
ectPool.G
etObject(
);
newButton
.transform
m.SetPare
nt(conten
tPanel);/
/se crean
los
botones
    en el
    panel
    "Content"
```

```
SampleButt
on
sampleButt
on =
newButton.
GetComponent
<SampleB
utton>();
sampleButt
on.GetComp
onentInChi
ldren<Text
>().text =
Dias[i];
```



```

52         sampleButton.name = Dias[i]; //copia el nombre de los splits "/"
53         n" (saltos de línea) encima de cada botón
54                                     /*
55                                     GameObject newButton2 =
56                                     buttonObjectPool2.GetObject();
57                                     newButton2.transform.SetParent
58                                     (contentPanelEliminar);
59                                     newButton2.name = PacientesB
60                                     [i];*/
61     }
62 }
63 void RemoveButtons() //para actualizar el numero de botones, se eliminan
64 todos y se vuelven a crear al darle al botón Refresh de la escena
65 Pacients.
66 {
67     while (contentPanel.childCount > 0)
68     {
69         GameObject toRemove = contentPanel.transform.GetChild
70         (0).gameObject;
71         buttonObjectPool1.ReturnObject(toRemove);
72         //GameObject toRemove2 = contentPanelEliminar.transform.GetChild
73         (0).gameObject;
74         //buttonObjectPool2.ReturnObject(toRemove2);
75     }
76 }
77 IEnumerator Temporizador() //Permite que a la iniciar la escena salga
78 automáticamente la lista de pacientes sin darle al boton refresh despues
79 de 0.55
80 {
81     yield return new WaitForSeconds(0.5f);
82     ButtonClick();
83 }
84 }

```

## SimpleObjectPool

```

1 using UnityEngine;
2 using System.Collections.Generic;
3 //Ubicado en Pacients, en el gameObject ButtonObjectPool, se utiliza para agrupar
  el prefab SampleButton y así el Shop Scroll list pueda hacer la invocación
  del botón.
4 // A very simple object pooling class
5 public class SimpleObjectPool : MonoBehaviour
6 {
7     // the prefab that this object pool returns instances of
8     public GameObject prefab;
9     // collection of currently inactive instances of the prefab
10    private Stack<GameObject> inactiveInstances = new Stack<GameObject>();
11
12    // Returns an instance of the prefab
13    public GameObject GetObject()
14    {
15        GameObject spawnedGameObject;
16
17        // if there is an inactive instance of the prefab ready to return,
        return that
18        if (inactiveInstances.Count > 0)
19        {
20            // remove the instance from the collection of inactive instances
21            spawnedGameObject = inactiveInstances.Pop(); 22
22        }
23        // otherwise, create a new instance
24        else
25        {
26            spawnedGameObject = (GameObject)GameObject.Instantiate(prefab);
27
28            // add the PooledObject component to the prefab so we know it came
            from this pool
29            PooledObject pooledObject =
                spawnedGameObject.AddComponent<PooledObject>();
30            pooledObject.pool = this; 31
31        }
32
33        // put the instance in the root of the scene and enable it
34        spawnedGameObject.transform.SetParent(null);
35        spawnedGameObject.SetActive(true);
36
37        // return a reference to the instance
38        return spawnedGameObject; 39
39    }
40
41    // Return an instance of the prefab to the pool
42    public void ReturnObject(GameObject toReturn)
43    {
44        PooledObject pooledObject = toReturn.GetComponent<PooledObject>();
45
46        // if the instance came from this pool, return it to the pool
47        if (pooledObject != null && pooledObject.pool == this)
48        {
49            // make the instance a child of this and disable it

```

```
50         toReturn.transform.SetParent(transform);  
51         toReturn.SetActive(false);
```

```
52
53         // add the instance to the collection of inactive instances
54         inactiveInstances.Push(toReturn);
55     }
56     // otherwise, just destroy it
57     /*else
58     {
59         Debug.LogWarning(toReturn.name + " was returned to a pool it wasn't ↗
           spawned from! Destroying.");
60         Destroy(toReturn);
61     }*/
62 }
63 }
64
65 // a component that simply identifies the pool that a GameObject came from
66 public class PooledObject : MonoBehaviour
67 {
68     public SimpleObjectPool pool;
69 }
```



**Vector4**

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 public static class Vector4
7 {
8     public static Quaternion ChangeQuat(this Windows.Kinect.Vector4 newvector,
9                                         Quaternion
10                                         newrotation)
11     {
12         return Quaternion.Inverse(newrotation) *
13             new Quaternion(-newvector.X, -newvector.Y, newvector.Z,
14                             newvector.W);
15     }
16 }
```

## A3.2 Codis PHP

### AngleInsert

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName = "Users";
7
8
9  //////////////////////////////////////
10 //
11 //////////////////////////////////////
12
13 //Variable from the patient
14 $Nombre = $_POST["DNIPost"];
15 $Angulo=$_POST["AnguloMaxPost"];
16 $Angulo2=$_POST["AnguloMax2Post"];
17 $Fecha= $_POST["FechaPost"];
18 $Hora=$_POST["HoraPost"];
19 $Movimiento=$_POST["MovimientoPost"];
20 $Simetria=$_POST["SimetriaPost"];
21 $RepHABDUP=$_POST["RepHABDUPPost"];
22 $HABDUptime=$_POST["HABDUptimePost"];
23 $Nivell=$_POST["NivellPost"];
24 $Bombtime=$_POST["BombtimePost"];
25
26
27
28
29
30 //Make Connection
31 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
32 //Check Connection
33 if(!$conn){
34     die("Connection Failed. ". mysqli_connect_error());
35 }
36 //
37
38 $sqc= "SELECT PatientName FROM patients WHERE
patients.DNI='".$Nombre."'"; 39
41 # $sql = "INSERT INTO angulos (DNIP, AngulomaxDerecha,
    AngulomaxIzquierda, PatientName, Dia, Hora,Ejercicio, Simetria)
    SELECT patients.DNI, '".
    $Angulo."', '". $Angulo2."', patients.PatientName, '". $Fecha."', '".
    $Hora."', '". $Movimiento."', '". $Simetria."' FROM patients
    WHERE patients.DNI='".$Nombre."'";
42 $sql = "INSERT INTO angulos (DNIP, AngulomaxDerecha,
    AngulomaxIzquierda, PatientName, Dia, Hora,Ejercicio, Simetria,
    RepeticionesHABDUP, TiempoHABDUP, Nivel, TiempoBomba) SELECT
    patients.DNI, '". $Angulo."', '".
    $Angulo2."', patients.PatientName, '". $Fecha."', '". $Hora."', '".
    $Movimiento."', '". $Simetria."', '". $RepHABDUP."', '". $HABDUptime."', '".
    $Nivell."', '". $Bombtime."' FROM patients WHERE patients.DNI='".$Nombre."'";
44 $result = mysqli_query($conn
176

```

```
, $sql);  
46     if(!result) echo "there was an error";
```

```
47     else echo "Everything  
    ok."; 48  
49 ?>
```

**DeletePatient**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
  "Users"; 7
8  ////////////////////////////////////////
   //
9  //Script que serve para Eliminar pacientes, a partir del script
   DeletePATient que está
10 //en el botón eliminar. Borra por lo tanto, todos los datos del paciente
   (del servidor) cuyo nombre es introducido
11 //y que corresponde al Doctor de ese paciente.
12 ////////////////////////////////////////
   //
13
14
15 //Variable from the patient
16 $PatientName = $_POST["PatientNamePost"];
17 $DoctorDNI=
  $_POST["DoctorNamePost"]; 18
19
20
21
22
23
24 //Make Connection
25 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
26 //Check Connection
27 if(!$conn){
28 die("Connection Failed. ".
mysqli_connect_error()); 29 }
30
31 $sql = "DELETE FROM `patients` WHERE patients.PatientName='".
  $PatientName.'"AND patients.DoctorDNI="'. $DoctorDNI.'"';
32 $result = mysqli_query($conn
, $sql); 33
34 if(!result) echo "there was an error";
35 else echo "Everything ok." , $PatientName, $DoctorName;
36 $conn-
>close() ; 37
38 ?>

```

**InsertPatient**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
  "Users"; 7
8
9  //////////////////////////////////////
10  //Recoge los datos insertados en el Registro de Pacientes mediante el
    script PatientInsert y los guarda en el servidor.
11  //////////////////////////////////////
12
13  //Variable from the patient
14  $PatientName = $_POST["PatientNamePost"];
15  $DNI= $_POST["PatientDNIPost"];
16  $DoctorDNI=
  $_POST["DNIPost"]; 17
18
19  //Make Connection
20  $conn = new mysqli($servername, $server_username, $server_password, $dbName);
21  //Check Connection
22  if(!$conn){
23  die("Connection Failed. ".
  mysqli_connect_error()); 24  }
25  //Inserta los datos en la tabla 'patients', creando un nuevo paciente,
    para un Doctor determinado
26
27  $sql = "INSERT INTO patients (PatientName,DNI,DoctorDNI)
28  VALUES ('".$PatientName."','".$DNI."','".$DoctorDNI."')";
29  $result = mysqli_query($conn
  , $sql); 30
31  if(!result) echo "there was an error";
32  else echo "Everything
  ok."; 33
34  ?>

```

**InsertUser**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
  "Users"; 7
8  //////////////////////////////////////
9  //Recoge los datos insertados en el Registro de fisioterapeuta mediante el
  script DataInserter y los guarda en el servidor.
10 //////////////////////////////////////
11 //Variable from the user
12 $FirstName = $_POST["FirstNamePost"];
13 $LastName = $_POST["LastNamePost"];
14 //$Age = $_POST["AgePost"];
15 //$Company = $_POST["CompanyPost"];
16 $DNI = $_POST["UserNamePost"];
17 // $Email = $_POST["EmailPost"];
18 $Password
  =$_POST["PasswordPost"]; 19
20
21
22
23 //Make Connection
24 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
25 //Check Connection
26 if(!$conn){
27   die("Connection Failed. ".
  mysqli_connect_error()); 28 }
29 //Inserta los datos en la tabla User, creando un nuevo
  usuario 30
31 $sql = "INSERT INTO user (FirstName,LastName,DNI>Password)
32   VALUES ('".$FirstName."','".$LastName."','".$DNI."','".$Password."')";
33 $result = mysqli_query($conn
  , $sql); 34
35 //if(!result) echo "there was an error";
36 echo "Usuario Registrado
  Correctamente"; 37
38 ?>

```

**listangulos**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
    "Users"; 7
8  ////////////////////////////////////////  ?
    //////////////////////////////////
9  //
10 ////////////////////////////////////////  ?
    //////////////////////////////////
11 //Variable from the patient
12 $PatientNamep = $_POST["PatientNamePost"];
13 $Dia =
    $_POST["DiaPost"]; 14
15
16 //Make Connection
17 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
18 //Check Connection
19 if(!$conn){
20     die("Connection Failed. ".
mysqli_connect_error()); 21 }
22
23 $sql = "SELECT Simetria,  ?
    Hora,Ejercicio,AngulomaxDerecha,AngulomaxIzquierda, Simetria FROM  ?
    `angulos` WHERE angulos.PatientNamep='".$PatientNamep.'" AND
    angulos.Dia='".$Dia.'"';
24 $result = mysqli_query($conn
, $sql); 25
26 //if(!result) echo "there was an error";
27 //else echo "";
28 if(mysqli_num_rows($result) > 0){
29     //show data for each row
30     while($row =
mysqli_fetch_assoc($result)) 31 {
32
33         echo "Ejercicio ".$row['Ejercicio']." \n\r Hora:  ?
            ".$row['Hora']." · Angulo Maximo Derecho:  ?
            ".$row['AngulomaxDerecha']." · Angulo Maximo Izquierdo:  ?
            ".$row['AngulomaxIzquierda']."\n\r
            Simetría:
            ".
            $row['Simetria']."\n\r_____ \n\r";
34     }
35 }
36
37 ?>

```



**Login**

```

1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbName = "Users";
6  $user_DNI=$_POST["UserNamePost"];
7  $user_Password=$_POST["PasswordPost"];
8  //////////////////////////////////////
9  //Script que, al insertar en el Login los datos de un usuario, comprueba
10 //si estos existen y si están bien (si la contraseña está bien). Si
11 //se vuelve al script "Login" del Unity y continua con el script( pasa a
12 //la escena "Pacients")
13 //////////////////////////////////////
14 //Make Connection
15 $conn = new mysqli($servername, $username, $password, $dbName);
16 //Check Connection
17 if(!$conn){
18     die("Connection Failed. ".
mysqli_connect_error()); 19 }
20
21
22 $sql = "SELECT Password FROM user WHERE DNI= ' ".$user_DNI." '";
23 $result = mysqli_query($conn
,$sql); 24
25
26 if(mysqli_num_rows($result) > 0){
27     //show data for each row
28     while($row = mysqli_fetch_assoc($result)){
29         if($row['Password']==$user_Password){
30             echo "Inicio de sesión correcto";
31         }
32         else{echo "Contraseña incorrecta, vuelva a
intentarlo";} 33 }
34     } else {
35         echo "Usuario no
encontrado"; 36     }
37
38
39
40
41
42 ?>

```



## LoginPaciente

```

1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbName = "Users";
6  $user_DNI=$_POST["DNIPost"];
7  //////////////////////////////////////
8  //Script que, al insertar en el Loginel DNI de un paciente, comprueba si
9  // este existe y si están bien (si la contraseña=DNI está bien). Si está
    bien,
10 //se vuelve al script "Login" del Unity y continua con el script( pasa a la
    escena "Pacients")
11 //////////////////////////////////////
12
13 //Make Connection
14 $conn = new mysqli($servername, $username, $password, $dbName);
15 //Check Connection
16 if(!$conn){
17     die("Connection Failed. ". mysqli_connect_error());
18 }
19
20
21 $sql = "SELECT DNI FROM `pacients`WHERE DNI= '". $user_DNI."'";
22 $result = mysqli_query($conn , $sql);
23
24
25 if(mysqli_num_rows($result) > 0){
26     //show data for each row
27     while($row = mysqli_fetch_assoc($result)){
28         if($row['DNI']==$user_DNI){
29             echo "Inicio de sesión correcto";
30         }
31         else{echo "Contraseña incorrecta, vuelva a intentarlo";}
32     }
33 } else {
34     echo "Usuario no encontrado";
35 }
36
37
38
39
40
41 ?>

```



**mostrardias**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
  "Users"; 7
8  //////////////////////////////////////  P
  //////////////////////////////////
9  //
10 //////////////////////////////////////  P
   //////////////////////////////////
11 //Variable from the patient
12 $PatientNamep =
$_POST["PatientNamePost"]; 13
14
15
16 //Make Connection
17 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
18 //Check Connection
19 if(!$conn){
20 die("Connection Failed. ".
mysqli_connect_error()); 21 }
22
23 $sql = "SELECT PatientNamep, Dia FROM `angulos` WHERE  P
angulos.PatientNamep='".$PatientNamep.'" GROUP BY Dia";//aquí muestra  P
solo los días q se ha hecho. Entonces
24 //yo lo que haria es aquí poner solo el día, nada de angulos y que cuando  P
se de a un botón (del día) salga la lista de angulos haciendo otro
script.
25
26 $result = mysqli_query($conn
,$sql); 27
28 //if(!result) echo "there was an error";
29 //else echo "";
30 if(mysqli_num_rows($result) > 0){
31 //show data for each row
32 echo "SELECCIONE EL DIA:\n\r";
33 while($row =
mysqli_fetch_assoc($result)) 34 {
35 echo "'".$row['Dia']."'.".\n\r";
36 }
37 }
38
39 ?>

```

**SeleccionDoctor**

```

1  <?php
2  //Variables for the connection
3  $servername = "localhost";
4  $server_username = "root";
5  $server_password = "";
6  $dbName =
  "Users"; 7
8  //////////////////////////////////////  ?
  //////////////////////////////////
9  //A partir del script Listado que se encuentra en la escena Pacientes  ?
  del Unity, este script
10 //({SeleccionDoctor.php) pide al servidor todos los nombres de pacientes de  ?
  un Doctor (DNI doctor) especifico
11 //////////////////////////////////////  ?
  //////////////////////////////////
12 //Variable from the patient
13 //$PatientName = $_POST["PatientNamePost"];
14 //$PatientAge= $_POST["PatientAgePost"];
15 $DoctorDNI=
$_POST["DNIPost"]; 16
17
18 //Make Connection
19 $conn = new mysqli($servername, $server_username, $server_password, $dbName);
20 //Check Connection
21 if(!$conn){
22 die("Connection Failed. ".
mysqli_connect_error()); 23 }
24
25 $sql = "SELECT PatientName FROM `patients` WHERE patients.DoctorDNI='".  ?
$DoctorDNI."'";
26
27 $result = mysqli_query($conn
,$sql); 28
29 //if(!result) echo "there was an error";
30 //else echo "";
31 if(mysqli_num_rows($result) > 0){
32 //show data for each row
33 echo "SELECCIONE SU PACIENTE:\n\r";
34 while($row =
mysqli_fetch_assoc($result)) 35 {
36 echo
"". $row['PatientName']. "\n\r"; 37 }
38 }
39
40 ?>

```